

УДК 004.451:004.658.2:004.43

DOI: 10.31673/2786-8362.2025.029186

Гордієнко К.О.; Ніщепенко Д.О.;
Герцюк М.М., PhD; Аронов А.О., к.т.н.;
Гавор А.С.

МАСШТАБОВАНІ ДЕЦЕНТРАЛІЗОВАНІ СИСТЕМИ НА ОСНОВІ РОЗПОДІЛЕНИХ СХОВИЩ ДАНИХ

Hordiienko K.O., Nishchemenko D.O., Gertsyuk M.M., Aronov A.O., Havor A.S. Scalable Decentralized Systems Based on Distributed Data Storage. The article presents an analysis of approaches to improving the scalability of decentralized systems through the integration of blockchain platforms with distributed NoSQL storage. The feasibility of using NoSQL databases as a distributed block storage layer within the ElasticBloC model is substantiated, which allows reducing the load on the blockchain and ensuring horizontal scalability. A system integration approach based on a multilayer software architecture using Java and Python is considered. As a result, a conceptual model of the synergistic combination of blockchain, NoSQL infrastructure, and application services is formulated, creating the prerequisites for building scalable and secure decentralized applications in DeFi environments. Approaches to integrating blockchain technologies with distributed NoSQL storage and architectural alignment of Java- and Python-based components within decentralized applications are systematized. The proposed solutions can be applied in the design of information systems with high requirements for resilience, reliability, and performance.

Keywords: blockchain, decentralized systems, distributed data storage, Java, microservices, NoSQL, object-oriented programming, Python

Гордієнко К.О., Ніщепенко Д.О., Герцюк М.М., Аронов А.О., Гавор А.С. Масштабовані децентралізовані системи на основі розподілених сховищ даних. У статті проведено аналіз можливостей підвищення масштабованості децентралізованих систем шляхом інтеграції блокчейн-платформ із розподіленими NoSQL-сховищами. Обґрунтовано доцільність використання NoSQL-баз даних у ролі розподіленого сховища блоків у моделі ElasticBloC, що дозволяє зменшити навантаження на блокчейн та забезпечити горизонтальне масштабування. Розглянуто підхід до системної інтеграції на основі багаторівневої програмної архітектури з використанням мов Java та Python. У результаті сформульовано концепцію синергетичного поєднання блокчейну, NoSQL-інфраструктури та прикладних сервісів, яка створює передумови для побудови масштабованих і захищених децентралізованих застосунків у середовищі DeFi. Систематизовано підходи до інтеграції блокчейн-технологій з розподіленими NoSQL-сховищами та обґрунтуванні концепції архітектурного поєднання Java- та Python-компонент у складі децентралізованих застосунків. Запропоновані положення можуть бути використані під час проектування інформаційних систем з високими вимогами до стійкості, надійності та продуктивності.

Ключові слова: блокчейн; децентралізовані системи; розподілені сховища даних; Java; мікросервіси; NoSQL; об'єктно-орієнтоване програмування; Python

Вступ

Сучасні інформаційні системи дедалі частіше орієнтуються на децентралізовані моделі оброблення даних, що зумовлено зростанням вимог до безпеки, стійкості та масштабованості цифрових сервісів. Блокчейн-технології забезпечують незмінність і прозорість транзакцій у децентралізованому середовищі, проте потребують інтеграції з високопродуктивними сховищами даних для подолання обмежень масштабованості. Одним із перспективних напрямів є поєднання блокчейну з NoSQL-архітектурами та використання мов програмування Java і Python для реалізації багаторівневих компонентів децентралізованих застосунків.

Постановка завдання. Попри активний розвиток децентралізованих систем, залишається невирішеним питання забезпечення їх масштабованості та продуктивності без втрати стійкості та криптографічної безпеки. Обмеження блокчейну на фізичному рівні зберігання та оброблення даних ускладнюють застосування технології у високонавантажених середовищах. NoSQL-системи мають потенціал компенсувати ці обмеження за рахунок горизонтального масштабування та розподіленої архітектури, однак потребують чітких підходів до інтеграції з блокчейн-платформами. Додатковим аспектом є вибір програмних засобів, здатних забезпечити узгоджену взаємодію компонентів у багаторівневій структурі децентралізованих

застосунків.

Аналіз останніх досліджень. У наукових публікаціях розглядаються питання організації консенсусу, забезпечення незмінності записів у блокчейні, застосування NoSQL-сховищ у розподілених системах, а також використання Java і Python для розробки компонентів блокчейн-орієнтованого програмного забезпечення. Однак більшість робіт фокусуються на окремих технологічних складових, не приділяючи достатньої уваги їх комплексній інтеграції у єдину архітектуру. Недостатньо висвітлено питання взаємодії блокчейну та NoSQL-сховищ на фізичному рівні, а також ролі мультипарадигмових мов програмування у забезпеченні гнучкості та масштабованості децентралізованих застосунків. Це визначає потребу систематизації існуючих підходів та узагальнення моделей їх спільного використання.

Метою роботи є аналіз підходів до інтеграції блокчейн-технологій, NoSQL-сховищ та багаторівневих програмних архітектур на основі Java і Python з метою підвищення масштабованості, ефективності та стійкості децентралізованих інформаційних систем.

Виклад основного матеріалу дослідження

Сучасна децентралізація спирається на технологію розподіленого реєстру (DLT), що забезпечує надійність, прозорість і незмінність транзакцій без центрального посередника, проте обмежена масштабованістю. Для подолання цього бар'єра блокчейн-реєстри інтегруються з NoSQL-сховищами, орієнтованими на горизонтальне масштабування. Для вирішення поставленого завдання необхідно послідовно розглянути ключові аспекти системної інтеграції – від вибору архітектурної моделі до поєднання технологічних компонентів і механізмів взаємодії даних. Так, архітектура ElasticBloC використовує HBase, побудовану на HDFS, як кластер масштабованого сховища блоків. У системній реалізації поєднуються можливості Java для створення корпоративних компонентів (наприклад, клієнтів консенсусу Besu/TeKu у мережах Proof-of-Stake) та Python – для тестування смарт-контрактів і побудови сполучних модулів, зокрема конектора BigchainDB-HBase. Така технологічна синергія формує основу високопродуктивних і безпечних децентралізованих застосунків (DApps) в екосистемі DeFi.

Блокчейн – це децентралізована, розподілена технологія реєстру (DLT), яка надійно реєструє транзакції через численні вузли. На відміну від традиційних баз даних, блокчейн не покладається на центральний орган, забезпечуючи незмінність та прозорість за допомогою криптографічного хешування та механізмів консенсусу. Ключові характеристики блокчейну:

- Децентралізація – жоден окремий суб'єкт не контролює весь реєстр.
- Незмінні записи – додані дані не можуть бути змінені чи видалені.
- Прозорість та безпека – транзакції є видимими та перевіряються всіма учасниками.
- Механізм консенсусу – транзакції вимагають мережевої згоди для підтвердження.
- Смарт-контракти – автоматизовані, самовиконувані контракти з попередньо визначеними умовами.

Традиційні бази даних (як-от реляційні СУБД та NoSQL бази даних) зберігають дані у централізованих системах, використовуючи архітектуру клієнт-сервер, де центральний орган керує доступом та безпекою.

Хоча блокчейн чудово підходить для безпеки та прозорості, традиційні бази даних все ще ефективніші для високошвидкісних транзакцій та модифікації даних.

Зі зростанням обсягу даних та користувачів, масштабованість та розподіл стали ключовими. NoSQL СУБД з'явилися як альтернативні системи управління даними, які часто не мають фіксованої схеми, не підтримують операції об'єднання і створені для горизонтального масштабування на базі реплікації та шардингу. Основні архітектури NoSQL включають кільцеві системи, master-slave та реплікаційні.

Таблиця 1

Порівняльна таблиця блокчейну та традиційних баз даних

Характеристика	Блокчейн	Реляційні бази даних	NOSQL
Структура даних	Розподілений реєстр (блоки, пов'язані криптографією)	Централізована, таблична	Схемонезалежна
Контроль	Децентралізований (peer-to-peer)	Централізований (адміністративний контроль)	Залежить від конкретної БД, в основному централізований
Незмінність	Записи постійні та захищені від несанкціонованого доступу	Дані можуть бути відредаговані або видалені	Дані можуть бути відредаговані або видалені
Швидкість/Продуктивність	Повільніша через процеси верифікації та консенсусу	Швидші транзакції та запити	Швидше виконання запитів
Масштабованість	Повільніша через механізми консенсусу	Обмежена вертикальна масштабованість	Велика горизонтальна масштабованість

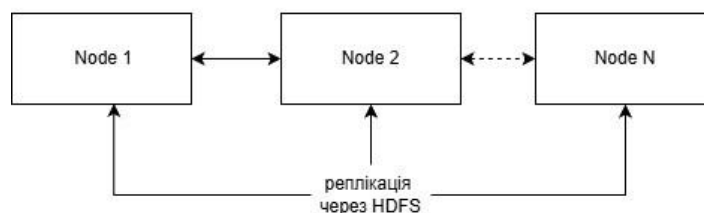


Рис. 1. Горизонтальне масштабування сховища блоків у кластері HBase/HDFS з реплікацією через HDFS

Однією з головних проблем блокчейну є обмежена масштабованість, особливо на фізичному рівні, що необхідно для зберігання зростаючого обсягу блоків. ElasticBloC – це масштабована архітектура, розроблена для приватних блокчейн-додатків, яка вирішує цю проблему, інтегруючи NoSQL-технології.

ElasticBloC використовує HBase як кластер сховища блоків. HBase, будучи колонково-орієнтованою базою даних, побудована на Hadoop Distributed File System (HDFS), що забезпечує масивну фізичну масштабованість для зберігання даних. HBase також забезпечує відмовостійкість, реплікуючи кожен блок на три вузли.

У розробці децентралізованих систем Java та Python займають важливі ніші, підтримуючи різні рівні архітектури DApps.

Java є надійною, безпечною та високопродуктивною мовою, яка є дружньою до корпоративного середовища. Її переваги активно використовуються у домені блокчейну:

- Розробка вузлів (нод): Можливо створювати та запускати локальні екземпляри вузлів блокчейну, повністю побудовані на Java. Наприклад, компанія LCHAIN розробляє мережу Hedera, яка є DLT-рішенням на основі Hashgraph, реалізованим на Java.
- Реалізація віртуальних машин: Java використовується для створення Java-базованих Ethereum Virtual Machine, що є "мозком" багатьох мереж, включаючи Ethereum та Hedera.
- NoSQL-системи: Такі системи, як Apache Cassandra та HBase, мають ключові модулі, написані на Java.

Крім того, існують Java SDK, такі як web3j Library, що можуть використовуватися для взаємодії з цілими DLT-мережами.

Python, відомий своєю читабельністю та високорівневою природою, також має значне місце у світі блокчейну.

- **Смарт-контракти:** Python може використовуватися для написання та тестування смарт-контрактів на блокчейн-платформах, таких як Xian Blockchain.
- **Розробка бекенду:** Python є критично важливим для комунікаційних та обробних компонентів децентралізованих архітектур. Наприклад, в архітектурі ElasticBloC (децентралізоване рішення для масштабованості) бекенд реалізовано за допомогою Python.
- **NoSQL-інтеграція:** Коннектор BigchainDB-HBase, який є ключовим компонентом ElasticBloC, що інтегрує блокчейн з масштабованим сховищем HBase (на базі HDFS), був реалізований на Python.

Таким чином, Java забезпечує стабільність та продуктивність ядра системи та корпоративну сумісність, тоді як Python використовується для гнучкої розробки, інтеграції та логіки взаємодії зі сховищами та API.

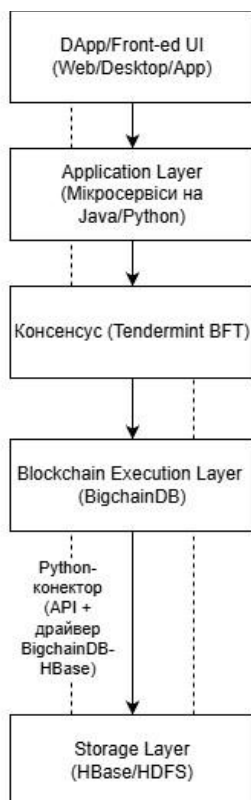


Рис. 2. Архітектурна взаємодія компонентів Dapps

Децентралізовані додатки (DApps) становлять ключову ланку екосистеми DeFi, забезпечуючи інтерфейс взаємодії між учасниками ринку. Попри те, що більшість сучасних DApps були створені у 2017-2019 роках, їх сфера застосування постійно розширюється.

Архітектура DApp є багатошаровою і складається із зовнішніх та внутрішніх частин. До внутрішніх частин належать блокчейн-платформа, смарт-контракти, токени, dApp-клієнти та інші платформи з якими DApp може взаємодіяти через API. До зовнішніх суб'єктів належать користувачі та розробники.

Для забезпечення консенсусу використовуються PoW, PoS і BFT, що визначають продуктивність і безпеку мереж.

Масштабованість блокчейну є критичним обмеженням. Для досягнення масивної масштабованості архітектури необхідне поєднання різних технологічних рішень, як на логічному, так і на фізичному рівні.

ElasticBloC – це архітектура, розроблена для забезпечення масштабованості блокчейн-додатків на фізичному рівні, використовуючи кластерні обчислення. Вона є прикладом

синергії NoSQL та блокчейну, реалізованої за допомогою Python та інтегрованої з технологіями BFT-консенсусу. ElasticBloC включає BigchainDB як блокчейн-рушій, Tendermint як механізм консенсусу на основі BFT і HBase/HDFS інтегровані за допомогою Python-конектора.

Ця архітектура демонструє, як NoSQL-системи, оптимізовані для великих наборів даних та горизонтального масштабування, можуть слугувати основою для фізичного масштабування блокчейн-реєстрів, усуваючи одну з головних його обмежень.

Децентралізація також активно впроваджується в архітектуру програмного забезпечення. Мікросервіси – це автономні, спеціалізовані служби, які працюють незалежно та спілкуються через API, уникаючи спільних баз даних. Це пріоритет децентралізації та простоти.

Мікросервіси дозволяють використовувати технологічну гетерогенність, що означає можливість використання різних мов, як-от Java, Python, Node.js, для різних служб. Це дозволяє розробникам обирати найкращий по сукупності характеристик інструмент для виконання того чим іншого завдання, зважаючи на загальну ефективність системи.

У мікросервісній архітектурі виникає проблема узгодженості даних, яку можна вирішити за допомогою шаблону Data Mesh (наприклад, використовуючи потокову передачу даних через Confluent Kafka). Такий підхід до децентралізованого управління даними відповідає філософії розподілених реєстрів.

У ході дослідження було систематизовано підходи до поєднання блокчейн-технологій із розподіленими NoSQL-сховищами та багаторівневими програмними архітектурами. Запропоновано концептуальну модель інтеграції, що передбачає використання HBase/HDFS для фізичного масштабування сховища блоків у межах архітектури ElasticBloC, а також розподіл функціонального навантаження між компонентами, реалізованими на Java та Python.

На основі аналізу існуючих рішень обґрунтовано доцільність використання Java для побудови високопродуктивних корпоративних елементів блокчейну (зокрема реалізацій вузлів, віртуальних машин і сервісів консенсусу) та Python – для задач інтеграції, тестування смарт-контрактів і побудови конекторів до розподілених сховищ. Запропонована схема взаємодії дозволяє забезпечити узгодженість даних між рівнями, мінімізувати затримки доступу та підвищити стійкість системи до навантажень.

Таке поєднання технологічних платформ дозволяє створити масштабовану архітектуру децентралізованих застосунків, яка забезпечує розширення обсягів транзакційних даних без втрати продуктивності та криптографічної цілісності. Зазначений підхід може бути використаний під час проектування інформаційних систем із високими вимогами до надійності, відмовостійкості та гнучкої підтримки розподілених структур даних.

Висновки

У роботі розглянуто підходи до підвищення масштабованості децентралізованих систем шляхом інтеграції блокчейн-платформ із розподіленими NoSQL-сховищами. Встановлено, що обмеження блокчейну на фізичному рівні зберігання даних та пропускну здатності ускладнюють використання технології в умовах значного транзакційного навантаження. Застосування NoSQL-систем, зокрема HBase, дозволяє забезпечити горизонтальне масштабування та розподілене зберігання великого обсягу блоків.

Показано доцільність використання різних мов програмування на різних рівнях архітектури децентралізованих застосунків: Java є базою для розробки високопродуктивних компонентів, пов'язаних з консенсусом та виконанням операцій блокчейну, тоді як Python доцільно застосовувати для компонентів інтеграції, тестування смарт-контрактів та організації доступу до розподілених сховищ.

Систематизовані підходи демонструють можливість створення архітектури, у якій блокчейн і NoSQL-сховище функціонують як взаємодоповнюючі елементи єдиної екосистеми, що забезпечує підвищення стійкості та масштабованості децентралізованих застосунків. Подальший розвиток досліджень може бути пов'язаний з оцінюванням продуктивності такої інтеграції в різних сценаріях навантаження, а також з удосконаленням механізмів взаємодії

між компонентами системи.

Список використаної літератури

1. Доценко О. Архітектура децентралізованих додатків у складі екосистеми defi. *Соціальна економіка*. 2022. № 64. С. 117–124. URL: <https://doi.org/10.26565/2524-2547-2022-64-10>.
2. 3rd global cryptoasset benchmarking study / A. Blandin et al. *SSRN electronic journal*. 2020. URL: <https://doi.org/10.2139/ssrn.3700822>.
3. A survey on Hadoop HBase system. *International journal of advance engineering and research development*. 2016. Vol. 3, no. 01. URL: <https://doi.org/10.21090/ijaerd.030114>.
4. Azzi R., Chamoun R. K., Sokhn M. The power of a blockchain-based supply chain. *Computers & industrial engineering*. 2019. T. 135. C. 582–592. URL: <https://doi.org/10.1016/j.cie.2019.06.042>.
5. Bakos Y., Halaburda H., Mueller-Bloch C. When permissioned blockchains deliver more decentralization than permissionless. *Communications of the ACM*. 2021. T. 64, № 2. C. 20–22. URL: <https://doi.org/10.1145/3442371>.
6. Bamakan S. M. H., Motavali A., Babaei Bondarti A. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert systems with applications*. 2020. T. 154. C. 113385. URL: <https://doi.org/10.1016/j.eswa.2020.113385>.
7. Blockbench / T. T. A. Dinh et al. *SIGMOD/PODS'17: international conference on management of data*, Chicago Illinois USA. New York, NY, USA, 2017. URL: <https://doi.org/10.1145/3035918.3064033>.
8. Casey M. J. In blockchain we trust. *MIT Technology Review*. URL: <https://www.technologyreview.com/2018/04/09/3066/in-blockchain-we-trust/>.
9. Chandrakant K., Piwowarek G. Implementing a simple Blockchain in Java. <https://www.baeldung.com>. URL: <https://www.baeldung.com/java-blockchain>.
10. Chen Y., Pereira I., Patel P. C. Decentralized governance of digital platforms. *Journal of management*. 2020. C. 014920632091675. URL: <https://doi.org/10.1177/0149206320916755>.
11. Davies B. How to build ethereum dapps: learn how to write smart contract code, design, implement, test, debug, deploy and secure a full end-to-end dapps. Independently Published, 2022.
12. Decision support for blockchain platform selection: three industry case studies / S. Farshidi et al. *IEEE transactions on engineering management*. 2020. T. 67, № 4. C. 1109–1128. URL: <https://doi.org/10.1109/tem.2019.2956897>.
13. Jibbawi H., Haque R., Ali Jaber Y. T. ElasticBloC: a massively scalable architecture for blockchain based applications. *Proceedings of the 2nd international conference on big data and cyber-security intelligence : CEUR Workshop Proceedings, Versailles, 16-17.12. 2019 / M.-R. Hojeij et al.* 2019. C. 73-82. URL: <https://ceur-ws.org/Vol-2622/paper11.pdf>.
14. Khobragade P., Turuk A. K. Blockchain consensus algorithms: a survey. *Lecture notes in networks and systems*. Cham, 2023. C. 198–210. URL: https://doi.org/10.1007/978-3-031-21229-1_19.
15. LSB: a lightweight scalable blockchain for iot security and anonymity / A. Dorri et al. *Journal of parallel and distributed computing*. 2019. T. 134. C. 180–197. URL: <https://doi.org/10.1016/j.jpdc.2019.08.005>.
16. The evolutionary trajectories of peer-produced artifacts: group composition, the trajectories' exploration, and the quality of artifacts / O. Arazy et al. *MIS quarterly*. 2020. Vol. 44, № 4. C. 2013–2053. URL: <https://doi.org/10.25300/misq/2020/15379>.
17. Understanding the motivations, challenges and needs of Blockchain software developers: a survey / A. Bosu et al. *Empirical software engineering*. 2019. Vol. 24, № 4. C. 2636–2673. URL: <https://doi.org/10.1007/s10664-019-09708-7>.

Автори статті

Гордієнко Катерина – старший викладач, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0002-5997-9682

Ніщенко Дмитро – старший викладач, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0006-1781-4109

Герцюк Микола – доктор філософії, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0000-0003-2946-9673

Аронов Андрій – кандидат технічних наук, доцент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0000-7868-8341

Гавор Артур – викладач, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0002-9705-1666

Authors of the article

Hordiienko Kateryna – senior lecturer, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0002-5997-9682

Nishchemenko Dmytro – senior lecturer, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0006-1781-4109

Gertsyuk Mykola – PhD (technical), Associate Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0000-0003-2946-9673

Aronov Andrii – Candidate of Sciences (technical), Associate Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0000-7868-8341

Havor Artur – lecturer, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0002-9705-1666