Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра
(освітній ступінь))

на тему: **Технології створення розподілених комп'ютерних систем зберігання даних на основі блокчейн**

Виконав: студент (ка) 6 курсу, групи СІм-61
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

	(підпис)	Гладій В.В. (прізвище та ініціали)
Керівник	(підпис)	Луцків А.М. (прізвище та ініціали)
Нормоконтроль	(підпис)	Луцик Н.С. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Осухівська Г.М. (прізвище та ініціали)
Рецензент	(підпис)	Стадник М.А. (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра комп'ютерних систем та мереж

ЗАТВЕРДЖУЮ
Завідувач кафедри *Осухівська Г.М.*

«_____» 2023 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *магістр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту *Гладію Віктору Васильовичу*
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) *Технології створення розподілених комп'ютерних систем зберігання даних на основі блокчейн*

Керівник проекту (роботи) *Луцків Андрій Мирославович, к.т.н., доц.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*01*» *грудня* 2023 року №*4/7-1132*

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи *Принципи організації розподілених систем, типи баз даних, характеристики блокчейн, можливості оптимізації розподілених систем*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз принципів і підходів до організації розподілених систем зберігання даних 2. Методи імплементації принципів технології блокчейн та оптимізації продуктивності виконання запитів у розподілених системах зберігання даних. 3. Апробація методів підвищення достовірності транзакцій та оптимальності виконання запитів у розподілених системах зберігання даних 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Архітектура розподілених баз даних. 4. Принципи технології блокчейн. 5. Критерії ефективності розподілених КС зберігання даних. 6. Метод оргназації РБД з блокчейн. 7. Результати експерименту. 8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та	*Осухівська Г.М., зав. каф. КІ*		
безпека в надзвичайних			
ситуаціях	*Стадник І.Я., проф. каф. ОХ*		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	*Аналіз принципів і підходів до організації розподілених*	*01.12.2023-*	*виконано*
	систем зберігання даних	*05.12.2023*	
2.	*Методи імплементації принципів технології блокчейн*	*05.12.2023-*	*виконано*
	та оптимізації продуктивності виконання запитів у	*12.12.2023*	
	розподілених системах зберігання даних		
3.	*Апробація методів підвищення достовірності*	*12.12.2023-*	*виконано*
	транзакцій та оптимальності виконання запитів у	*17.12.2023*	
	розподілених системах зберігання даних		
4.	*Охорона праці та безпека в надзвичайних ситуаціях*	*18.12.2023*	*виконано*
5.	*Оформлення пояснювальної записки*	*20.12.2023*	*виконано*
6.	*Оформлення графічного матеріалу*	*21.12.2023*	*виконано*
7.	*Попередній захист кваліфікаційної роботи магістра*	*22.12.2023*	*виконано*
8.	*Захист кваліфікаційної роботи магістра*		

Студент _____ *Гладій В.В.*
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ *Луцків А.М.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Технології створення розподілених комп’ютерних систем зберігання даних на основі блокчейн // Кваліфікаційна робота магістра// Гладій Віктор Васильович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2023 // с. – 82, рис. – 38 , табл. –10 , аркушів А1 –8 , додат. – 1, бібліогр. – 21.

Ключові слова: технологія, розподілена система, дані, зберігання, блокчейн.

У кваліфікаційній роботі магістра проаналізовано класи розподілених систем зберігання даних та визначено, що основними з них є гомогенні та гетерогенні системи, які відрізняються як на рівні типів використовуваного апаратного, так і програмного забезпечення. Проведено аналітичний огляд архітектур розподілених систем зберігання даних та визначено рівні їх організації, зокрема концептуальний, зовнішній та внутрішній, які дали змогу оцінити можливість оптимізації в контексті виконання розподілених транзакцій.

Запропоновано метод імплементації технології блокчейн для організації класичних розподілених систем зберігання даних шляхом додавання до кожної таблиці бази даних кортежу атрибутів: часова мітка, цифровий підпис попередньої транзакції, цифровий підпис поточної транзакції, публічний ключ користувача та булевого поля щодо операції видалення. Розроблено метод оптимізації виконання запитів до розподілених систем зберігання даних, який заснований на формуванні одного та багатьох моментальних знімків бази даних з оптимальним їх розташуванням за рахунок кластеризації подібних знімків, що дало змогу підвищити у 50 разів продуктивність опрацювання запитів.

ABSTRACT

Technologies for creating distributed computer systems for data storage based on blockchain /Master thesis / Hladii Victor / Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm -61 // Ternopil, 2023// p. - 82, fig. – 38, table. – 15, Sheets A1 – 8, Add – 1, Ref. – 21.

Keywords: technology, distributed system, data, storage, block chain.

The master's qualification work analyzed the classes of distributed data storage systems and determined that the main ones are homogeneous and heterogeneous systems, which differ both in terms of the types of hardware and software used. An analytical review of the architectures of distributed data storage systems was carried out and the levels of their organization, in particular conceptual, external and internal, were determined, which made it possible to assess the possibility of optimization in the context of the execution of distributed transactions.

A method of implementing blockchain technology for the organization of classical distributed data storage systems is proposed by adding to each database table a tuple of attributes: time stamp, digital signature of the previous transaction, digital signature of the current transaction, public key of the user and a Boolean field regarding the deletion operation. A method of optimizing the execution of requests to distributed data storage systems was developed, which is based on the formation of one and many snapshots of the database with their optimal location due to the clustering of similar snapshots, which made it possible to increase the performance of query processing by 50 times.

ЗМІСТ

ВСТУП

Актуальність теми. Розподілені бази даних (РБД) відіграють вирішальну роль в організації сучасних обчислень, особливо в контексті великих даних, забезпечення властивостей масштабованості та відмовостійкості. Це дає змогу будувати високопродуктивні комп'ютерні системи різного призначення, починаючи від системи моніторингу на основі IoT, закінчуючи системами керування космічними польотами.

Розподілені БД забезпечують можливість доволі просто і легко розширювати можливості зберігання та опрацювання даних. Замість того, щоб покладатися на один монолітний сервер бази даних, дані можна розподіляти між кількома вузлами або серверами. Це дає змогу опрацьовувати великі об'єми даних і задовольняти підвищений попит користувачів без жодної точки збою.

Забезпечення високої доступності та відмовостійкості комп'ютерних систем з розподіленими базами даних реалізується за допомогою реплікації. При цьому дані дублюються між кількома вузлами, тому, якщо один вузол виходить з ладу, доступ до даних усе ще можна отримати з інших вузлів. Це забезпечує високу доступність і мінімізує час простою, що є критичним для додатків, які повинні працювати 24/7. Розподіляючи інформацію та їх обробку між кількома вузлами, БД часто можна забезпечити кращу продуктивність, ніж одна централізована база даних. При цьому дані опрацьовуються паралельно із скороченням часу запитів і покращенням загальної швидкості реагування системи.

РБД часто містять вбудовані механізми балансування навантаження, які рівномірно розподіляють навантаження запитів і транзакцій між вузлами. Це запобігає тому, що будь-який окремий вузол стане вузьким місцем продуктивності. Реплікація даних у розподілених базах даних гарантує, що дані є надлишковими та можуть бути відновлені у разі їх

втрати або апаратних збоїв. Це усуває необхідність у складних процесах резервного копіювання та відновлення.

Хоча налаштування та керування системою розподіленої бази даних може бути складним процесом, однак це також може бути економічно ефективним з точки зору апаратних ресурсів. Це в свою чергу передбачає використання звичайного апаратного забезпечення з можливістю масштабування лише за потреби, уникаючи потреби у дорогих високоякісних серверах. Розподілені бази даних часто підтримують різні моделі даних, включаючи реляційні, NoSQL і NewSQL, що робить їх придатними для різних типів програм і вимог до даних.

Таким чином, розподілені бази даних важливі для забезпечення здатності масштабування, забезпечення високої доступності, відмовостійкості і продуктивність. Вони є фундаментальною технологією для вирішення проблем, пов’язаних із великими даними, і вимогами сучасних додатків із інтенсивним об’ємом даних.

Дослідженню принципів побудови та організації розподілених баз даних присвячено багато наукових праць як українських так і закордонних вчених. Зокрема, щодо моделей і принципів опрацювання розподілених баз даних багато праць присвячено такими вченим як Пасічник В.В., Резніченко В.А., Берко А. Ю., Верес О. М., Трофіменко О.Г.. Серед закордонних науковців дослідженням способів організації та ефективного опрацювання розподілених даних займались такі вчені як P. Valduriez, M. Tamer Özsu, Saeed K. Rahimi, Frank S. Haug та ін.

Однак, для більшості сучасних розподілених баз даних існує проблема забезпечення високої надійності доступу до даних та їх зберігання. Як альтернативу класичним реляційним та нереляційним розподіленим базам даних можна розглянути технологію блокчейн (block chain). Прямо використовувати її у якості розподіленої БД неможливо, оскільки в повній мірі не розроблено сигнатуру та методи доступу до даних. Тому актуальною науково-технічною задачею на сьогодні є дослідження методів і способів

організації розподілених баз даних на основі технології блокчейн, що дозволить значно підвищити ефективність та захищеність даних при побудові складних комп’ютерних систем.

Мета кваліфікаційної роботи полягає у дослідженні теоретичного і прикладного застосування блокчейн технологій при організації комп’ютерних систем з розподіленими базами даних .

Для досягнення вказаної мети у роботі поставлено наступні **задачі:**

– аналіз наукових праць і прикладних досліджень щодо організації розподілених баз даних;

– аналітичний огляд підходів щодо проектування комп’ютерних систем на основі блокчейн технологій та можливостей організації відповідних розподілених БД;

– обґрунтування та вдосконалення математичного забезпечення побудови РБД на основі блокчейн технології;

– розробка методу організації РБД з використанням блокчейн технології;

– програмна реалізація розподіленої БД на основі блокчейн технології.

Об’єкт дослідження: процес організації та маніпулювання даними у розподілених базах даних.

Предмет дослідження: методи і засоби побудови розподілених баз даних з використанням технології блокчейн.

Методи дослідження: Для вирішення поставлених задач використано наступні методи: аналіз та узагальнення – при проведенні аналізу існуючих до організації розподілених систем зберігання даних; формалізації – при розробці методів забезпечення достовірності транзакцій та оптимізації розміщення моментальних знімків БД; проектування та програмування – при розробці бази даних з імплементованими елементами технології блокчейн; експеримент та вимірювання – при перевірці достовірності результатів запропонованого методу.

Наукова новизна отриманих результатів. Наукова новизна полягає у вирішенні науково-практичної задачі імплементації технології блокчейн в організацію розподілених систем зберігання даних.

– уперше запропоновано метод імплементації технології блокчейн для організації класичних розподілених систем зберігання даних шляхом додавання до кожної таблиці бази даних кортежу атрибутів: часова мітка, цифровий підпис попередньої транзакції, цифровий підпис поточної транзакції, публічний ключ користувача та булевого поля щодо операції видалення, що дало змогу забезпечити достовірність та можливість виявлення несанкціонованого внесення змін у дані незалежно від ролі користувача;

– уперше запропоновано метод оптимізації виконання запитів до розподілених систем зберігання даних, який заснований на формуванні одного та багатьох моментальних знімків бази даних з оптимальним їх розташуванням за рахунок кластеризації подібних знімків, що дало змогу підвищити у 50 разів продуктивність опрацювання запитів.

Практичне значення одержаних результатів. Проведено експериментальні дослідження щодо практичної реорганізації розподілених реляційних баз даних з використанням оптимізації на основі кластерного аналізу моментальних знімків БД, що дало змогу значно підвищити (у 50 разів) пропускну здатність опрацювання запитів та забезпечити оптимальність розміщення моментальних знімків (з відхиленням до 2%) за вузлами мережі у порівнянні з точними оптимальними розміщеннями

Публікації. Результати кваліфікаційної роботи апробовані на XII Міжнародній науково-практичній конференції молодих учених та студентів (6-7 грудня 2023 р.) та XI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року) як тези конференцій.

1. Луцків А.М., Гладій В.В. Особливості функціонування та класифікації розподілених систем зберігання даних. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 455.

2. Луцків А.М., Гладій В.В. Структура та взаємодія між блоками у блокчейн. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 145.

Структура роботи. Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 81 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1
АНАЛІЗ ПРИНЦИПІВ І ПІДХОДІВ ДО ОРГАНІЗАЦІЇ РОЗПОДІЛЕНИХ СИСТЕМ ЗБЕРІГАННЯ ДАНИХ

1.1. Аналіз основних понять при організації класичних розподілених систем зберігання даних

Під розподіленою базою даних, зазвичай, розуміють систему, яка застосовується для управління процесом зберігання і пошуку даних у кількох взаємопов’язаних базах даних [1]. У цьому випадку взаємопов’язані бази даних в переважній більшості розташовані в різних географічних локаціях.

У розподілених базах даних можна прозоро отримувати доступ і зберігати інформацію у різних територіально-розподілених місцях і забезпечити високу доступність, масштабованість і механізми відмовостійкості.

РБД розроблені для опрацювання величезних об’ємів даних, які генеруються різними і ,зазвичай, неоднорідними системами. Такий підхід може забезпечити безперебійну роботу та високу продуктивність при обміні даними та комунікації між організаціями або її частинами [1].

Існує деяка сукупність функцій, які роблять розподілені бази даних дуже популярним при структуруванні та організованому зберіганні інформації.

Фрагментація – загальна система бази даних розділена на менші підмножини, які є її фрагментами. При цьому можливе використання одного з трьох видів фрагментації – горизонтальна (розділена за рядками на основі предикатів), вертикальна (розділена за стовпцями в залежності від накладених умов) і гібридна (горизонтальна + вертикальна) [1].

Реплікація даних – у цьому випадку, територіально-розподілена база даних підтримує та зберігає декілька копій однієї і тієї ж інформації в

окремих фрагментах, щоб забезпечити властивості доступності даних, відмовостійкості і високої продуктивності [1].

Стратегія розподілу даних – визначає, чи потрібно зберігати всі фрагменти даних у передбачених серверах і використовується для зниження мережевого трафіку та оптимізації продуктивності [1].

Прозорість даних – властивість, що полягає у здатності РСКБД приховувати від користувачів усі процедури і складнощі та забезпечувати їм прозорий доступ до необхідних даних і програм [1].

Функції управління розподіленою базою даних покладено на централізовану підсистему, яка забезпечує керування базою даних таким чином, що складається враження наче уся інформація зберігається в одній локації.

До основних функцій і властивостей системи керування РБД належить [1]:

– створення, отримання, оновлення та видалення баз даних;

– періодична синхронізація сховища даних і забезпечення механізмів доступу, завдяки яким розповсюдження стає прозорим для користувачів, що гарантує повне оновлення даних, змінених у будь-якому місці;

– застосування у прикладних областях, де наявні великі обсяги даних, які необхідно опрацювати із забезпеченням одночсансого доступу багатьох користувачів;

– підтримка неоднорідних платформ, на яких функціонують бази даних;

– підтримка та забезпечення конфіденційності і цілісності інформації у баз даних.

До основних факторів, які стимулюють перехід та необхідність застосування РБД належать [1]:

– Особливості організаційної структури підприємства − сьогодні більшість компаній є територіально-розподіленими як в межах окремої

країни, так і окремих континентів. При цьому кожен організаційний підрозділ потребує власної множини локальних даних, що породжує необхідність організації розподілених фрагментів бази даних.

– Необхідність обміну даними в межах різних підрозділів – різні структурні елементи компанії потребують комунікації, що супроводжується обміном даними та іншими інформаційними ресурсами. Як наслідок, це спричиняє необхідність формування спільних фрагментів бази даних або синхронізованих реплік.

– Підтримка OLTP та OLAP – онлайн опрацювання транзакцій (OLTP) та онлайн аналітичне опрацювання (OLAP) працюють у різноманітних системах, які можуть володіти спільними даними. РСКБД підтримує опрацювання обох цих видів інформаційних ресурсів, надаючи синхронізовані дані.

– Регенерація (відновлення) бази даних – одна з найбільш поширених процедур, яка застосовується до РСКБД є реплікація у різних локаціях зберігання даних. Реплікація в автоматичному режимі забезпечує відновлення даних у випадку, коли вони були пошкодженими у якомусь з вузлів системи. Кінцеві користувачі системи мають можливість одержати доступ до даних з інших працездатних локацій БД, поки пошкоджений фрагмент розподіленої системи реконструюється. Отже, збій або вихід з ладу фрагменту бази даних може стати практично непомітним для користувачів.

– Підтримка кількох прикладних програм − більшість організацій використовують різноманітні прикладні застосунки, кожен з яких взаємодіє зі «своєю» базою даних, а РСКБД бере на себе забезпечення однакової функціональності при маніпулюванні одними і тими ж даних на різних платформах.

До переваг РСКБД відносно централізованих СКБД належить [1]:

– підтримка гнучкості модульної розробки комп’ютерних систем;

– підвищена надійність функціонування БД;

– висока швидкість надання відповідей на запити користувачів;
– низька вартість обміну даними.

Гнучкість модульної розробки при застосуванні підходу РБД полягає у здатності забезпечити розширюваність місць зберігання та опрацювання даних. На відмінну від централізованих систем, даний процес є менш трудомістким і не порушує надійності працездатності комп'ютерної системи в цілому. При організації РБД така задача передбачає інтеграцію нових робочих станцій і відповідних локалізованих даних з подальшим їх підключенням до вже існуючої системи без внесення змін, які передбачають перебої у функціональності системи.

Підвищена надійність РБД проявляється у випадку, коли відбувається збій у її роботі. При цьому користувачі все ж мають можливість використовувати та опрацьовувати дані, однак з дещо меншою продуктивністю. У випадку застосування централізованих систем зберігання та опрацювання даних, збій у роботі БД приводить до повної зупинки працездатності комп'ютерної системи.

Перевага у швидкості і точності видачі відповідей на користувацькі запити до РСКБД пов'язана з тим, що у випадку ефективної організації БД, відповіді (response) можуть надсилатися з локальних вузлів. Це дає змогу пришвидшити формування відгуків системи в цілому. На противагу РБД, у централізованих системах усі запити проходять через єдиний вузол управління, що підвищує час видачі відповіді на запит користувача.

Низька вартість передачі та обміну даними проявляється у випадку, коли дані РБД, які зберігаються локально, використовуються також локальними користувачами. При цьому значно знижуються комунікаційні витрати при маніпулюванні інформацією, що неможливо при організації централізованих систем.

Однак, поряд з важливими перевагами, якими володіють РБД, існує і ряд недоліків. Основними з них є [1]:

– доволі великий бюджет і складність побудови БД;

– додаткова вартість, спричинена забезпеченням ефективності маніпулювання даними;

– додаткові витрати у випадку не оптимального розподілу даних;

– складність забезпечення цілісності даних.

Висока вартість і складність побудови РБД проявляється, у першу чергу, у необхідності використання комплексного і дороговартісного ПЗ, що реалізує властивість прозорого доступу до даних та координації локацій їхнього зберігання.

Додаткові витрати, пов’язані з маніпулюванням даними в РБД, проявляються у випадку, коли при виконанні доволі простих операцій існує велика кількість відношень і виконується надмірна кількість обчислень з метою забезпечення синхронізації даних у всіх вузлах розподіленої системи.

Додаткові витрати при не оптимальному розподілі даних пов’язані із низькою продуктивністю при формуванні відповідей на запити користувачів.

Складність забезпечення цілісності даних при організації РБД зумовлена необхідністю забезпечення актуальності даних у місцях її розподілу.

1.2. Особливості класифікації розподілених баз даних

Загалом, розподілені бази даних можна класифікувати за середовищем реалізації та функціонуванням на гомогенні (однорідні) та гетерогенні (неоднорідні). Структуру такої класифікації показано на рис 1.1.

У гомогенних РБД усі вузли, між якими розподілені дані, використовують одні і ті ж СКБД та ОС. Основними властивостями однорідних РСКБД є:

– вузли розподіленої системи використовують подібне або однакове програмне забезпечення, зокрема це стосується СКБД;

– кожен вузол системи володіє інформацією про інші вузли та комунікує з ними при опрацювання користувацьких запитів;

– забезпечення доступу до даних виконується через спільний інтерфейс, що емулює роботу з єдиною БД.

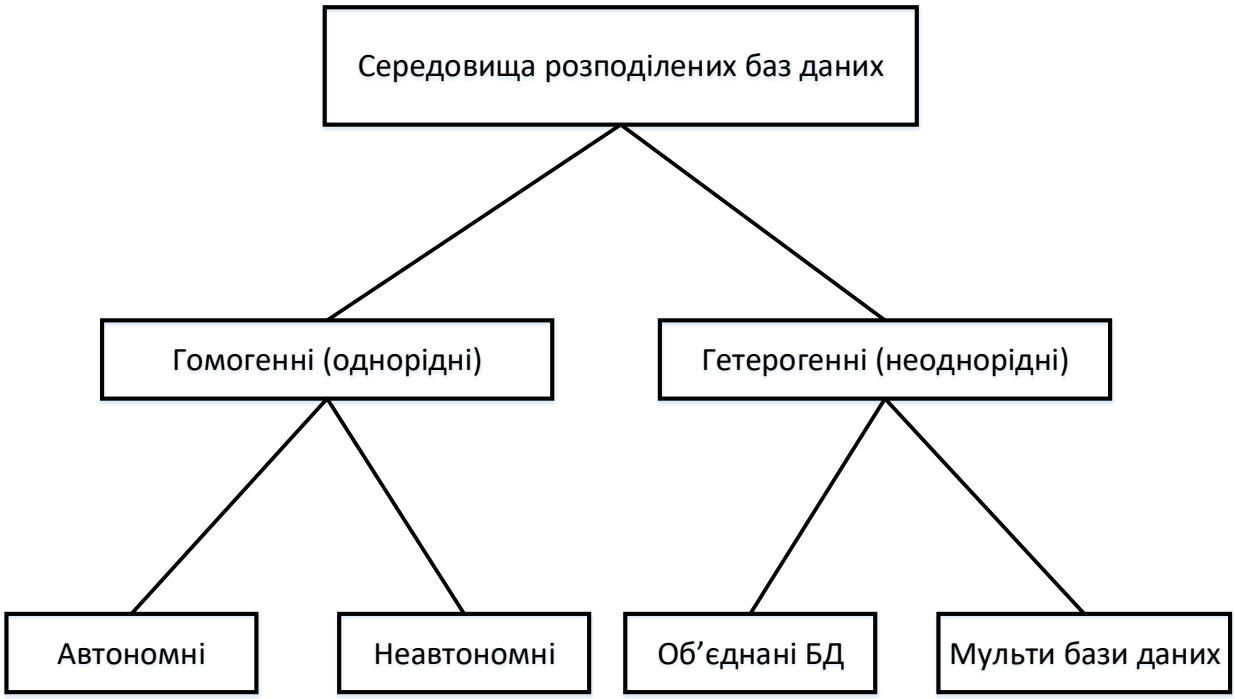

Рис. 1.1. Види середовищ розподілених баз даних

Існує два різновиди гомогенних РБД: автономна та неавтономна.

У випадку автономної РБД, кожна БД, що входить до її складу функціонує незалежно від інших. Інтеграцію таких БД забезпечує зовнішня програмна система управління передачею повідомлень при обміні даними і їх станами.

Неавтономна РБД передбачає розподіл даних між гомогенними вузлами, а центральна СКБД координує оновлення даних у вузлах розподілу.

У неоднорідній РБД на різних вузлах встановлені різні операційні системи, використовуються різні СКБД з різними моделями даних.

До основних властивостей гетерогенних РБД належать:

– різні вузли містять різні схеми БД та прикладне ПЗ, тобто до складу системи можуть входити різні СКБД, наприклад, реляційні, об’єктно-орієнтовані, документо-орієнтовані та ін.;

– опрацювання запитів ускладнене у зв’язку з різними схемами організації даних;

– опрацювання транзакцій вимагає значних ресурсів як апаратних, так і програмних;

– вузли можуть не знати про фрагменти розподіленої системи, тому комунікація при опрацюванні користувацьких запитів є обмеженою.

Підвидами неоднорідних РБД є об’єднанні БД та мульти бази даних.

Для об’єднаних РБД характерним є їх незалежність (за своєю природою) та водночас інтеграція у єдину комп’ютерну систему.

Мульти БД передбачають наявність і застосування центрального модуля координації дій та доступу до баз даних.

1.3. Аналіз архітектур розподілених баз даних

Архітектури РБД зазвичай проектуються з врахуванням трьох параметрів:

– розподіл – вказує на фізичний розподіл даних між різними вузлами;

– автономність – вказує на розподіл контролю над системою бази даних і ступінь, до якого кожна складова СКБД може працювати незалежно.

– неоднорідність – стосується однорідності або відмінності моделей даних, системних компонентів і баз даних.

Деякі з поширених архітектурних моделей [2]:

– клієнт - серверна архітектура;

– однорангова мережева архітектура;

– мультиархітектура СКБД.

Клієнт - серверна архітектура представляє собою дворівневу організацію, де функціональність розділена на сервери та клієнти. Функції сервера передусім охоплюють керування даними, обробку запитів, оптимізацію та керування транзакціями.

Клієнтські функції включають переважно інтерфейс користувача. Однак вони мають деякі функції, такі як перевірка узгодженості та керування транзакціями.

Існує два різні способи організації архітектури клієнт-сервер. Перший передбачає у розподіленій системі один сервер і кілька клієнтів, другий – кілька серверів і кілька клієнтів (рис. 1.2).

При організації розподіленої бази даних на основі моделі однорангової мережі, кожен вузол системи функціонує одночасно як клієнт, і як сервер, забезпечуючи при цьому сервіси бази даних.

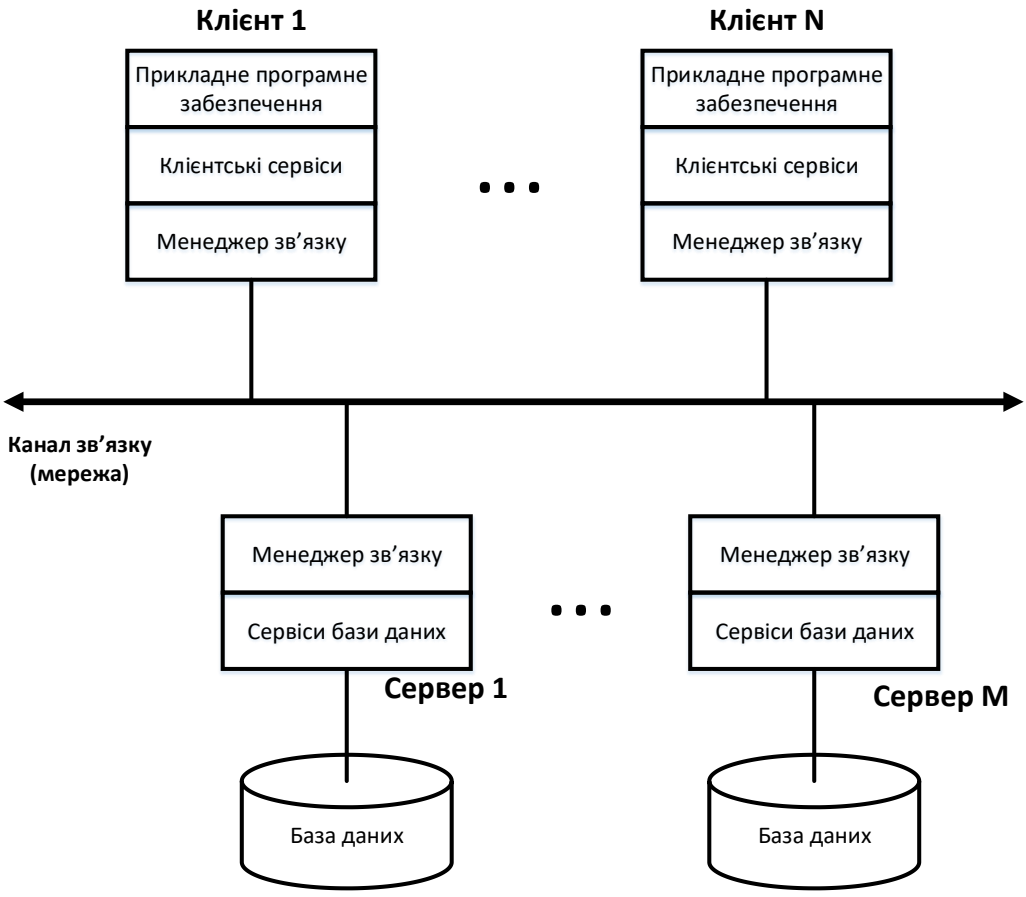

Рис. 1.2. Клієнт-серверна модель організації розподіленої комп'ютерної системи з багатьма клієнтами та багатьма серверами

Усі вузли за умови організації у вигляді однорангової мережі є рівнозначними. Цією архітектурою передбачено чотири рівні схем [2]:

– глобальна концептуальна схема – відображає глобальний логічний розподіл та представлення даних;

– локальна концептуальна схема – відображає логічну організацію даних на кожному вузлі системи;

– локальна внутрішня схема – відображає фізичну організацію даних на кожному вузлі;

– зовнішня схема – відображає дані у зручному для користувачів вигляді.

На рис. 1.3 показано архітектуру організації РБД на основі однорангової мережі.

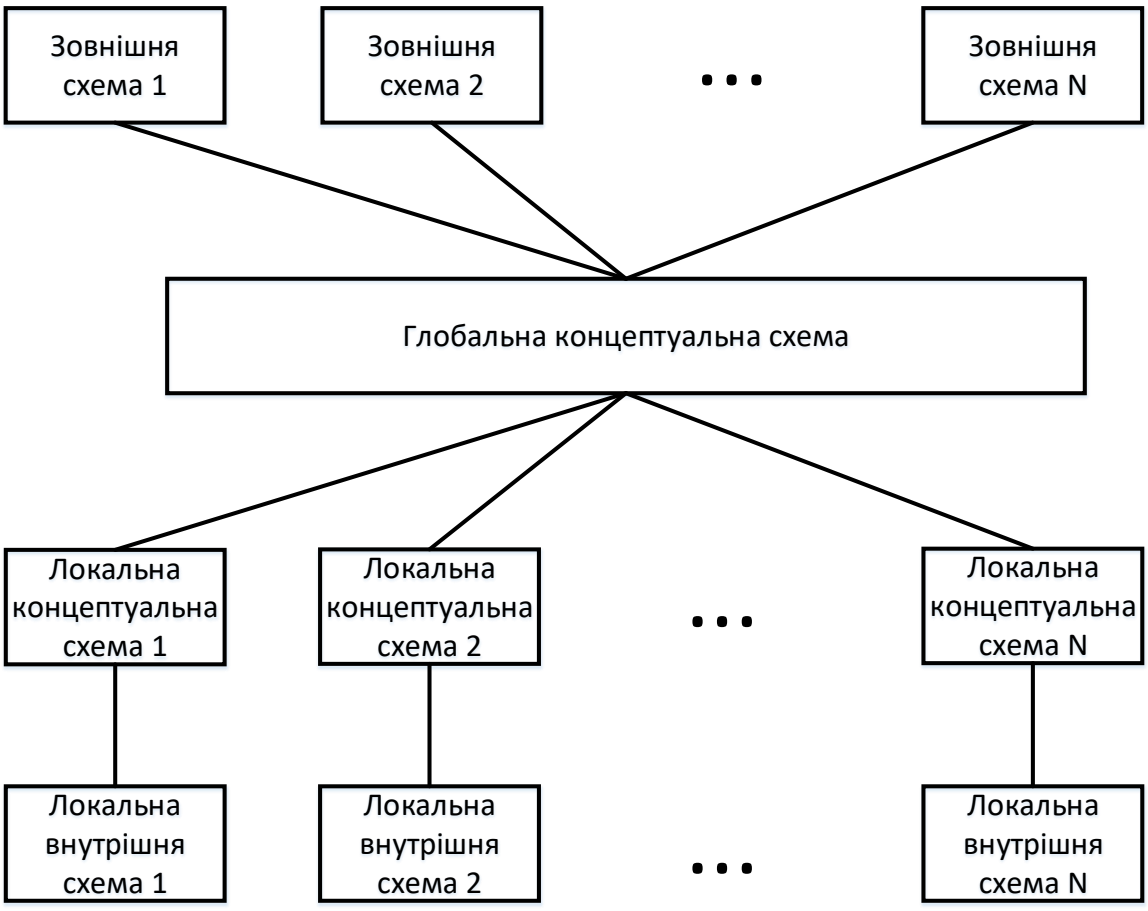

Рис. 1.3. Архітектура РБД на основі однорангової мережі

Мультиархітектури СКБД представляють собою інтегровані системи баз даних, утворених сукупністю двох або більше автономних систем баз даних.

Мульти СКБД можна зобразити за допомогою шести рівнів та відповідних схем [2]:

– зовнішній рівень кількох баз даних – відображає декілька інтерфейсів користувачів, що складаються з підмножин інтегрованої розподіленої бази даних;

– концептуальний рівень кількох баз даних – описує інтегровану мультибазу даних, яка складається з глобальних логічних визначень структури кількох баз даних;

– внутрішній рівень кількох баз даних – відображає розподіл даних між різними вузлами та зіставлення кількох баз даних із локальними даними;

– зовнішній рівень локальної бази даних − відображає загальнодоступний перегляд локальних даних;

– концептуальний рівень локальної бази даних − описує локальну організацію даних на кожному сайті;

– внутрішній рівень локальної бази даних – відображає фізичну організацію даних на кожному вузлі.

Існує два варіанти проектування мульти СКБД [2]:

– модель із концептуальним рівнем кількох баз даних;

– модель без концептуального рівня з кількома базами даних.

На рис. 1.4 та рис. 1.5 показано модель архітектури розподілених мульти баз даних з наявним концептуальним рівнем та без нього відповідно.

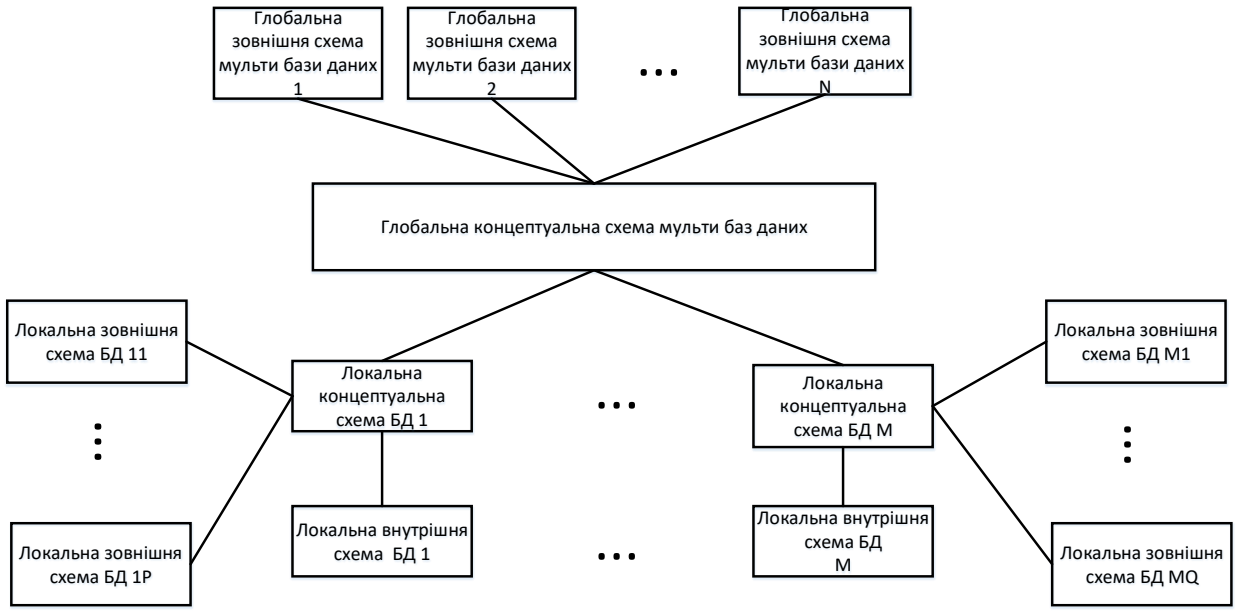

Рис. 1.4. Модель архітектури мульти баз даних з глобальним концептуальним рівнем

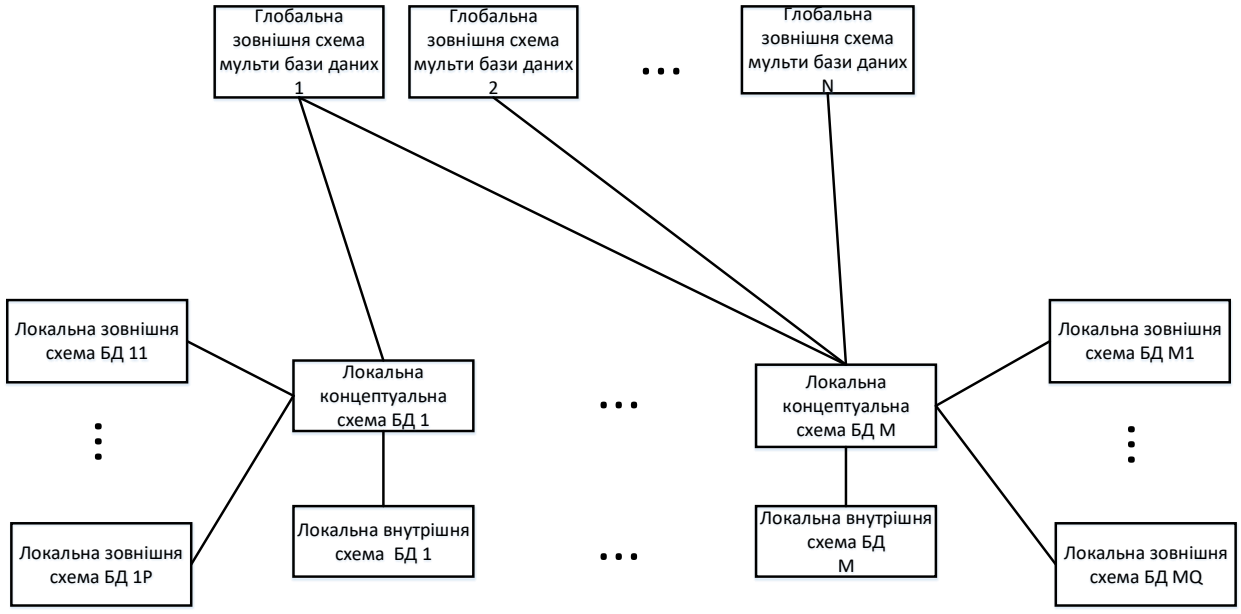

Рис. 1.5. Модель архітектури розподілених мульти баз даних без глобального концептуального рівня

Альтернативами при дизайні розподілу таблиць у РБД визначають наступні [2]:

- реляційні відношення нерепліковані і нефрагментовані;
- повністю репліковані таблиці;
- частково репліковані відношення;
- фрагментовані відношення;
- змішаний розподіл.

При проектуванні розподілених БД без використання реплікації та фрагментації різні реляційні відношення містяться у різних вузлах системи. При цьому дані зберігаються у безпосередній близькості від місця, де вони використовуються найчастіше. Такий підхід найбільше підходить для систем баз даних, де низький відсоток запитів, необхідних для об'єднання інформації в таблицях, розміщених на різних вузлах. Якщо прийнято відповідну стратегію розподілу, то ця альтернатива дизайну допомагає зменшити вартість зв'язку під час обробки даних [2].

При використанні альтернативи дизайну із застосуванням повної реплікації, на кожному вузлі зберігається одна копія всіх таблиць бази даних. Оскільки кожен вузол системи має власну копію всієї бази даних, запити виконуються дуже швидко, що не потребує значних витрат на передачу даних. Навпаки, величезна надлишковість даних вимагає великих витрат під час операцій оновлення і синхронізації даних. Отже, повна реплікація підходить для систем, де потрібна обробка великої кількості запитів, тоді як кількість оновлень бази даних невелика.

При частковій реплікації копії таблиць або частини таблиць зберігаються у різних вузлах розподіленої системи [3]. Розподіл таблиць здійснюється відповідно до частоти доступу. При цьому враховується той факт, що частота доступу до таблиць значно відрізняється між вузлами. Кількість копій таблиць (або частин) залежить від того, як часто виконуються запити доступу та вузла, який генерує запити доступу.

Якщо при побудові розподілених систем використовується підхід фрагментації, то це передбачає її поділ на кілька частин, які називаються фрагментами або розділами, і кожен з них може зберігатися на різних

вузлах. Це враховує той факт, що рідко буває так, що всі дані, які зберігаються у таблиці, потрібні на даному вузлі [3].

Крім того, фрагментація збільшує паралелізм і забезпечує краще аварійне відновлення. Тут у системі є лише одна копія кожного фрагмента, тобто немає зайвих даних. Існує три техніки фрагментації: вертикальна, горизонтальна і гібридна.

При проектуванні розподіленої комп'ютерної системи за допомогою змішаних підходів відбувається поєднання технік фрагментації та часткових реплікацій. Тут таблиці спочатку фрагментуються у будь-якій формі (горизонтальній або вертикальній), а потім ці фрагменти частково тиражуються на різних вузлах відповідно до частоти доступу до фрагментів.

1.4. Висновки до розділу

У даному розділі одержано наступні результати:

1 Проведено аналіз основних понять, якими оперують при організації розподілених систем зберігання даних, визначено переваги і недоліки застосування таких систем, що дало змогу визначити шляхи оптимізації при формуванні та виконанні транзакцій з великим об'ємом даних.

2 Проаналізовано класи розподілених систем зберігання даних та визначено, що основними з них є гомогенні та гетерогенні системи, які відрізняються як на рівні типів використовуваного апаратного, так і програмного забезпечення.

3 Проведено аналітичний огляд архітектур розподілених систем зберігання даних та визначено рівні їх організації, зокрема концептуальний, зовнішній та внутрішній, які дали змогу оцінити можливість оптимізації в контексті виконання розподілених транзакцій.

РОЗДІЛ 2
МЕТОДИ ІМПЛЕМЕНТАЦІЇ ПРИНЦИПІВ ТЕХНОЛОГІЇ БЛОКЧЕЙН ТА ОПТИМІЗАЦІЇ ПРОДУКТИВНОСТІ ВИКОНАННЯ ЗАПИТІВ У РОЗПОДІЛЕНИХ СИСТЕМАХ ЗБЕРІГАННЯ ДАНИХ

2.1. Особливості технології блокчейн та визначення шляхів її імплементації у розподілених базах даних

Технологія блокчейн використовує різні математичні концепції і алгоритми для забезпечення безпеки, цілісності та функціонування. Основні математичні аспекти технології блокчейн включають:

1. Криптографія – криптографічні методи відіграють важливу роль у блокчейні:

- хеш-функції – використовуються для перетворення даних в унікальний хеш-код, що забезпечує створення блоків і забезпечення цілісності даних.
- публічний та приватний ключі – застосовуються при створенні криптографічних підписів, які підтверджують автентичність та авторство транзакцій.
- симетричне та асиметричне шифрування: Використовуються для захисту конфіденційності даних та забезпечення доступу до них лише власникам приватних ключів.

2. Доказ роботи (Proof of Work, PoW) і доказ відбору (Proof of Stake, PoS) – алгоритми, які базуються на математичних концепціях, що дозволяють мережі блокчейн досягати консенсусу та визначати правила створення нових блоків.

3. Криптографічні хеш-функції для генерації адрес і хешів блоків – дозволяють створювати унікальні адреси для користувачів та генерувати ідентифікатори для блоків і транзакцій.

4. Розподіленість і консенсусні алгоритми – технологія блокчейн використовує математичні принципи, щоб досягти консенсусу між вузлами мережі і дозволити їм спільно підтверджувати і реєструвати транзакції.

5. Смарт-контракти – програми, що виконуються в середовищі блокчейн, можуть використовувати математичні операції та логіку для автоматизації угод та операцій на мережі.

6. Математична теорія графів – деякі аспекти блокчейну використовують теорію графів для моделювання зв'язків між блоками та транзакціями в ланцюжку.

Застосування цих математичних концепцій дозволяє блокчейну забезпечити безпеку, надійність, цілісність даних і довести консенсус між всіма учасниками мережі без потреби довіри до центральної авторитетної сторони.

Актуальність розробки розподілених баз даних на основі блокчейну полягає в кількох ключових аспектах:

1. Безпека та надійність – блокчейн надає високий рівень безпеки і надійності завдяки своїм криптографічним методам, безперервному моніторингу всіх транзакцій і неможливості зміни чи видалення інформації, що особливо важливо в критичних областях, таких як фінанси, медицина та ланцюжки постачання.

2. Децентралізація – розподілені бази даних на основі блокчейну дозволяють відмовитися від потреби спеціалізованих централізованих серверів, що дає змогу зменшити ризик відмови комп’ютерної системи через відмову одного центрального вузла і робить систему більш стійкою до атак.

3. Прозорість та аудит – технологія блокчейн дозволяє створювати розподілені системи, які надають велику прозорість угод та операцій. Будучи розподіленою, публічною базою даних, блокчейн може бути перевіреним ким завгодно, що збільшує рівень довіри до системи.

4. Смарт-контракти – блокчейн може бути використаним для реалізації смарт-контрактів, які автоматизують і створюють угоди. Це може відкривати нові можливості для автоматизації бізнес-процесів і зменшення ризиків.

5. Доступність – розподілені бази даних на основі блокчейну можуть бути доступні з будь-якого місця і в будь-який час, що особливо важливо для організацій, які мають глобальний характер та потребують гарантованого доступу до даних.

6. Безпека даних – завдяки криптографії та механізмам контролю доступу, блокчейн може забезпечити високий рівень захисту конфіденційної інформації.

7. Низькі витрати на інфраструктуру – зменшення потреби в централізованих інфраструктурах та управлінні дозволяє знизити витрати на інфраструктуру.

8. Нові бізнес-моделі – технологія блокчейн може дозволити створити нові бізнес-моделі, особливо в областях, де довіра та безпека мають високий статус.

Однак важливо відзначити, що блокчейн не є універсальним рішенням, і його використання має бути обґрунтованим в конкретному випадку.

Технологія має свої обмеження, такі як обмежена масштабованість, високі витрати на опрацювання даних (особливо в публічних блокчейнах), і потребує великих обчислювальних ресурсів. Тому перед впровадженням блокчейну слід ретельно аналізувати його призначення та вигоди.

Так, технологію блокчейн можна використовувати для створення розподілених і безпечних баз даних, які часто називають «базами даних блокчейну» або «розподіленими базами даних на основі блокчейну».

Технологія блокчейн має унікальні особливості, завдяки яким вона добре підходить для певних випадків використання, коли безпека даних, незмінність, прозорість і децентралізація є критично важливими.

Наведемо кілька прикладів, коли блокчейн може сприяти створенню безпечних і розподілених баз даних.

Незмінність даних – дані, що зберігаються в блокчейні, є незмінними, тобто після запису транзакції в блок її неможливо змінити або видалити. Ця незмінність забезпечує цілісність даних і запобігає несанкціонованому втручанню, що робить його придатним для програм, де безпека даних є першочерговою.

Децентралізація – блокчейни працюють у децентралізованій мережі вузлів, де кожен вузол підтримує копію всього блокчейну. Така децентралізація усуває окремі точки відмови, підвищує відмовостійкість і гарантує, що жодна юридична особа не має повного контролю над даними.

Прозорість і можливість перевірки: прозорий характер блокчейна дозволяє будь-кому, хто має доступ до мережі, перевіряти та перевіряти транзакції. Ця прозорість може мати вирішальне значення для додатків, які вимагають високого рівня підзвітності та довіри, таких як управління ланцюгом поставок і фінансові операції.

Безпека: блокчейни використовують криптографічні методи для захисту даних і транзакцій. Публічні та дозволені блокчейни відомі своїми надійними функціями безпеки, що робить їх стійкими до різних типів кібератак.

Розподілений консенсус: блокчейни використовують механізми консенсусу, такі як підтвердження роботи (PoW) або підтвердження частки (PoS), для підтвердження та узгодження порядку транзакцій. Цей розподілений консенсус забезпечує точність і послідовність даних у мережі.

Смарт-контракти: багато блокчейн-платформ підтримують смарт-контракти, які є самовиконуваними, програмованими контрактами, які автоматично виконують заздалегідь визначені дії, коли виконуються певні умови. Розумні контракти можуть сприяти складній взаємодії та автоматизувати процеси децентралізованої поведінки.

Конфіденційність даних: Деякі платформи блокчейну пропонують функції конфіденційності, що дозволяє зберігати конфіденційні дані в мережі безпечним і приватним способом. Це особливо цінно для програм, які вимагають конфіденційності даних. Хоча технологія блокчейн пропонує значні переваги для певних випадків використання, важливо також враховувати її обмеження, включаючи масштабованість, енергоспоживання (у випадку блокчейнів на основі PoW), а також складність розробки та обслуговування.

Бази даних Blockchain особливо добре підходять для таких додатків, як управління ланцюгом поставок, медичні записи, цифрова ідентифікація, фінансові транзакції та будь-які випадки використання, які вимагають довіри, безпеки та децентралізованого контролю. Однак вони можуть бути не найкращим рішенням для кожного сценарію, і їх застосування слід ретельно оцінювати на основі конкретних вимог проекту.

2.2. Формалізація задачі підвищення безпеки та масштабованості розподілених систем зберігання даних

Математичне забезпечення при організації розподілених комп’ютерних систем зберігання даних варто будувати з урахуванням кращих практик і нотацій, які характерні для технології блокчейн. Для формування задачі щодо побудови моделі розподілених БД з використанням блокчейну варто розглянути наступний приклад

Нехай існує деяка реляційна база даних, яка використовується для зберігання замовлень різних клієнтів і постачальників. З часом клієнти та постачальники можуть оновлювати дані: додавати нові замовлення, вносити зміни до попередніх або видаляти існуючі. Довіра є надзвичайно важливим аспектом управління електронними даними. Система повинна підтримувати походження всіх транзакцій бази даних при виникненні конфліктних ситуацій.

Для прикладу, клієнт C_1 може стверджувати, що замовлення було скасовано до узгодженого терміну, але постачальник S_2 може оскаржити час скасування. Таких сценаріїв на практиці існує досить багато.

Коли виникають такі розбіжності щодо історії бази даних, необхідно мати ефективний процес аудиту, для якого висновок є незаперечним і, отже, задовольняє усі сторони. Для вирішення задачі побудови такого аудиту потрібно розробити надійну реляційну систему бази даних, побудовану на існуючій та зрілій технології (наприклад, сучасному стандарті RDBMS1), щоб зберігати транзакції таким чином, щоб уся історія бази даних була захищеною від підробки або підміни. Зокрема, хоча зараз не існує можливості запобігти небажаним модифікаціям бази даних (наприклад, зловмисним користувачам root або випадковим збоям обладнання), будь-яка модифікація даних та їх історії буде виявлена під час процесу аудиту.

Іншими словами, довіряти стану бази даних можна лише у тому випадку, якщо переконатись, що вона є цілісною та не містила втручань ззовні. База даних також має підтримувати ефективну оцінку аналітичних запитів. Передбачається, що багато користувачів можуть надсилати до бази даних велику кількість запитів, які часто називають робочим навантаженням.

Кожен запит перевіряє стан бази даних на певну часову мітку запиту. Кожен запит може бути простим запитом на вибірку або таким, що містить складні перетворення та агрегації даних. Таким чином, ще однією функціональною вимогою до бази даних є підтримка навантажень запитів у великому масштабі.

Нехай D – реляційна база даних. У цьому випадку інтерес викликає її еволюція, тобто база даних повинна бути параметризована за допомогою критерію версії ($version$).

Стан бази даних у деякій версії задається $D(t)$, де t – часова мітка. Для простоти можна припустити, що $t \in \mathbb{N}^+$. Нехай U – користувачі бази даних. Транзакція – це зміна, яка застосовується до бази даних (додавання,

видалення та модифікація кортежів у D і позначається як ΔD. Позначення оновленої бази даних після виконання транзакції ΔD_t, надісланої користувачем $u \in U$, як:

$$D(t+1) = D(t) + \Delta D_t \tag{2.1}$$

де $D(t+1)$ – наступний стан бази даних після попереднього $D(t)$;

ΔD_t – транзакція, що перевела базу даних у стан $D(t+1)$.

Припускаючи, що база даних починається з порожнього початкового стану, $\emptyset$, при $t = 0$, можна визначити часову шкалу бази даних до деякого часу t у вигляді ряду $TL(t)$, який можна задати формулою:

$$TL(t) = \begin{bmatrix} \emptyset \\ null \\ 0 \end{bmatrix}, \begin{bmatrix} D_1 \\ u_1 \\ 1 \end{bmatrix}, \begin{bmatrix} D_2 \\ u_2 \\ 2 \end{bmatrix} \dots \begin{bmatrix} D_{t-1} \\ u_{t-1} \\ t-1 \end{bmatrix} \tag{2.2}$$

де D_i — і-та транзакція, надіслана користувачем u_i у момент часу $t = i$. Далі необхідно розробити системну БД для керування TL шкали часу з такими властивостями:

– отримання шкали часу: для будь-якого часу $t > 0$ шкалу часу $TL(t)$ можна повністю отримати з БД.

– отримання знімка: для будь-якого часу $t > 0$ стан бази даних D можна відновити з БД

– перевірка: існує така функція верифікації – $DB \longmapsto \{true, false\}$, яка виконує перевірку системи таким чином, що будь-яка підроблена версія системної $DB' \neq DB$ не пройде перевірку. Не існує жодних припущень щодо контролю доступу до сховища, тому джерелом втручання може бути рівень root доступу до БД. Таким чином, накладається обмеження щодо проблеми виявленням, а не запобіганням втручанню.

– робоче навантаження запиту: нехай q буде деяким запитом, визначеним на знімку $D(t_q)$, де t_q є міткою часу запиту. Системна БД може ефективно оцінювати велику кількість таких запитів із різними часовими мітками запиту. Набір запитів називається робочим навантаженням запиту і може бути записаний у вигляді – $Q = \{q_1, q_2, \dots, q_N\}$.

Останніми роками питання достовірності привертає увагу як науковців, так і практиків у різних областях, включаючи розподілені системи реального часу, схеми обміну токенами в хмарі і використання файлів журналізації для криміналістичного аналізу.

У кваліфікаційній роботі основна увага зосереджена на дослідженні достовірності транзакцій у середовищі, де користувачам не можна довіряти, навіть таким, які мають повний доступ до основної бази даних.

Сценарій привілейованих зловмисних користувачів обговорювався кількома дослідниками [3, 14, 13]. Використовувані підходи були засновані на мережі та перевірці даних механізмів виявлення фальсифікації бази даних. На відміну від цього, пропонується покладатися на вбудовані блокчейни, які не потребуватимуть глибокого доступу до рівня системного рівня (наприклад, активність файлової системи або перевірка мережевого трафіку).

Існують також дослідження у галузі криміналістичної інспекції на рівні бази даних з використанням тригерів і відповідних функцій бази даних. При масштабній роботі тригери можуть бути непомірно дорогими. На відміну від підходу базованого на тригерах, пропонується використовувати ефективну спеціальну перевірку структур блокчейну для визначення цілісності бази даних.

Відповіді на запити з використанням матеріалізованих представлень є загальноприйнятим полем у базі даних. Основна увага була зосереджена на представленнях, які визначаються загальними запитами SQL. У пропонованому підході до організації розподілених систем зберігання даних, знімки генеруються запитами у дуже специфічній формі, і, таким

чином, можна специфікувати проблему вибору матеріалізованого перегляду, щоб отримати більш ефективний алгоритм для пошуку оптимального рішення.

2.3. Формальний опис процесу проектування розподілених комп'ютерних систем зберігання даних на основі блокчейн

Базуючись на формальному означенні проблеми, яка наведена вище, необхідно описати та формалізувати процес проектування розподілених баз даних на рівні системи керування та з врахуванням особливостей технології блокчейн. При такому проектуванні пропонується інтегрувати блокчейни у реляційні таблиці для підтримки властивості незмінності транзакцій і перевірки.

2.3.1. Користувачі, ключі та цифрові підписи

Нехай U – сукупність користувачів. Кожен екземпляр в U може бути користувачем у традиційному розумінні, мобільним пристроєм або пристроєм інтернету речей (IoT). Без втрати загальності їх усіх вважають користувачами розподіленої бази даних.

Користувач $u \in U$ однозначно характеризується парою відкритих і закритих ключів: public(u), private(u). Пари ключів, зазвичай, генеруються за схемою RSA з використанням великої кількості бітів (наприклад, 4096), щоб уникнути колізій ключів, навіть якщо в U є мільярди екземплярів.

Для довільних даних x функція кодування задається як: $\big(x, private(u)\big) \longmapsto \hat{x}$, де $\hat{x}$ є кодуванням без втрат x, широко відомим як зашифрований текст.

Зашифрований текст можна знову розшифрувати за допомогою функції кодування, але з відкритим ключем:

$$enc\big(\hat{x}, public(u)\big) = x \tag{2.3}$$

Щоб підтвердити авторство деяких даних x користувачем u, використовується така схема цифрового підпису:

1. Розрахунок хешу x за допомогою фіксованої хеш-функції – $h(x)$.
2. Обчислення зашифрованого тексту $h(x)$ – $sig(x) = enc(h(x), private(u))$.
3. Публікація даних, підпису і відкритого ключа: $\langle x, sig(x), public(u)\rangle$.

Опубліковані дані не розголошують приватний ключ користувача. Щоб перевірити достовірність авторства за опублікованими даними необхідно виконати наступні кроки:

1. Розшифрування підпису – $y = enc(sig(x), public(u))$.
2. Переконатися, що декодований підпис є хеш-значенням x за допомогою фіксованої хеш-функції – $h(x) \overset{m}{=} y$, де $m =?$.

2.3.2. Блокчейн у реляційних таблицях

У роботі пропонується схема доповнення кожної реляційної таблиці в базі даних D додатковими атрибутами для зберігання часової мітки транзакції, підписом нової таблиці, підписом попереднього стану таблиці та відкритого ключа користувача. Для цього також потрібен додатковий бітовий прапорець, щоб вказати, чи є транзакція видаленням.

На практиці бітовий прапорець видалення може бути доповнений до кінця підпису транзакції, але для зручності читання припускається, що до таблиці додано ще один логічний атрибут.

Для кожної таблиці T в базі даних D з атрибутами $attr(T)$ робиться припущення, що принаймні один атрибут, id, є унікальним ідентифікатором рядка.

Після цього таблиця T доповнюється додатковими атрибутами: $(t, sig, sig', pubkey, del?)$, кожен відповідає описаному вище. Нова таблиця з доповненими атрибутами записується як $\hat{T}$. Тут потрібно звернути увагу на те, що відстежується два підписи: sig та sig', що відповідають підписам

таблиці після $T(t)$ і перед $T(t\ -\ 1)$ транзакцією відповідно. Це необхідно для забезпечення цілісності блокчейну.

2.3.3. Оновлення та верифікація транзакцій

Припустимо, що користувач $u\ \in\ U$ хоче здійснити транзакцію ΔD. Для кожної таблиці T, яка бере участь у транзакції, нехай ΔT буде рядками, на які це впливає (додано, змінено або видалено).

Для кожного кортежу $x \in \Delta T$ виконується операція вставки у $\hat{T}$ розширеного кортежа $\langle x, i, s, s', public(u), del?\,(x)\rangle$, де i — мітка часу поточної транзакції, s, s' – підписи як описано нижче, public(u) – це відкритий ключ автора транзакції, і, нарешті, $del?\,(x)$ – це логічний прапорець, який вказує, чи видалено x у транзакції.

Цифрові підписи представляються наступним чином:

– $s' = x[sig'] : x[t\} = i - 1$ – однозначно визначається як підпис попередньої транзакції з міткою часу $i\ -\ 1$.

– $s = enc(hash(\Delta T + s'), private(u))$ – цифровий підпис даних транзакції та попередній підпис.

Верифікація транзакції передбачає, що кожна транзакція ΔT на рівні таблиці однозначно ідентифікується своєю міткою часу t. Вона також унікально ідентифікується своїм підписом.

Перевірка транзакції формально представляється наступним чином:

$$verify(\Delta \mathrm{T}) = enc\big(sig_t, public(u)\big) \stackrel{\text{m}}{=} hash(\Delta T + sig_{t-1}) \quad (2.4)$$

Це дозволяє верифікувати всі таблиці $\hat{T}$ у тимчасовій базі даних. Будь-яке втручання буде виявлено функцією перевірки.

2.4. Забезпечення оптимальності відповіді на запити до БД

Оптимальність відгуку розподіленої системи зберігання даних на запит користувача передбачає знаходження необхідного знімку бази даних і надання достовірної інформації за найкоротший проміжок часу. Тому актуальним є дослідження робочого навантаження при роботі з базою даних.

Кожна таблиця $\hat{T}$ є часовою шкалою транзакцій, підписаних цифровим підписом і доступних для перевірки.

Щоб підтримувати та відповідати на довільний запит q, потрібно обчислити знімки всіх таблиць $\hat{T}$, які потрібні q у момент часу t_q. Це означає, що необхідно виконати операцію групування за допомогою ідентифікатора рядка (з вихідної таблиці T), при цьому кожна група агрегується до рядка з останньою часовою міткою транзакції.

Після цього виконується видалення усіх рядків з логічними прапорцями видалення, встановленими на true. Таку процедуру можна виразити за допомогою SQL-запитів, оскільки вони підтримують функції для роботи з вікном, як показано на рис 2.1.

snapshot($\hat{T}, t_q$):

```
WITH V AS (
   SELECT id, {last_value(x) : x ∈ attr(T)} OVER W
   FROM T̂
   WHERE T̂.t ≤ t_q
   WINDOW W AS PARTITION BY id ORDER BY T̂.t
)   SELECT id, {x : x ∈ attr(T)}
FROM V
WHERE NOT V.del?
```

Рис. 2.1. Формальний запис запиту агрегації знімку БД за запитом у конкретний момент часу

Таким чином, запит q до розподіленої системи зберігання даних, обчислюється як $q(snapshot(\widehat{T_1}, t_q), snapshot(\widehat{T_2}, t_q) \dots)$.

Наступні кроки щодо імплементації технології блокчейн у традиційні розподілені системи зберігання даних полягають у забезпеченні і дослідженні сімейства методів оптимізації для ефективного опрацювання робочого навантаження запитів $Q = \{q_1, q_2, \dots, q_N\}$ шляхом оптимального вибору фіксованої кількості знімків.

2.5. Забезпечення оптимальності при формуванні єдиного знімку бази даних при відповіді на запит користувачів

Якщо існує знімок $S = snapshot(T, t_s)$ і запит q з часовою міткою запиту t_q/t_s, то можна можемо оцінити q, побудувавши моментальний знімок (T, t_q) на основі S. Це передбачає застосування всіх транзакцій в інтервалі $[t_s, t_q]$ (транзакції фіксації) або $[t_q, t_s]$ (транзакції відкочування).

Оскільки кожна транзакція має свою унікальну мітку часу, кількість транзакцій в інтервалі визначається як $|t_q - t_s|$.

Припустимо, що наявний лише один моментальний знімок s і робоче навантаження Q запиту. Коли кожен запит $q \in Q$ використовує s для обчислення знімка (S, t_q), то одержують модель витрат, яка представлена у формулі (2.5).

$$cost(Q|s) = \sum_{q \in Q} |t_q - t_s| \qquad (2.5)$$

Це призводить до першої проблеми оптимізації, яку можна озачити наступним чином:

Означення 1 (проблема оптимального розміщення одного знімка): «Задано робоче навантаження запиту і необхідно знайти таке оптимальне розміщення знімка s^*, де $s^* = \underset{s}{\operatorname{argmin}}\, \operatorname{cost}(Q|s)$».

Можна показати, що розміщення окремого знімка може бути вирішено, розмістивши знімок на медіані часових міток запиту. Для цього використовується теорема розміщення одного моментального знімка, яка представляється формулою:

$$t_s^* = median(t_q : q \in Q) \tag{2.6}$$

Формування кількох миттєвих знімків дає змогу підвищити продуктивність за допомогою кількох моментальних знімків. Це приводить до виникнення проблеми щодо оптимальності формування кількох знімків таблиць бази даних.

Нехай $S = \{s_1, s_2, \ldots, s_m\}$ і Q робоче навантаження запиту. Кожен запит буде оцінено за допомогою найближчого знімка. Отже, вартість тоді можна розрахувати за формулою:

$$cost(Q|S) = \sum_{q \in Q} min\{|t_q - t_s| : s \in S\} \tag{2.7}$$

Означення 2 (проблема розміщення оптимальних багатьох знімків). Нехай $S^{**} = \{s_1, s_2, \ldots, s_m\}$ множина m знімків. Тоді задача оптимальності формулюється так: «Які є розміщення моментальних знімків так, щоб вартість $(Q|S)$ була мінімізованою?

Позначимо S^{**} як $opt(Q, m)$. При цьому варто зауважити, що проблема багатьох моментальних знімків має властивість, яку називають оптимальністю підзадач. У зв'язку з цим формулюється теорема щодо оптимальності підзадач при формуванні багатьох знімків.

Нехай $S^* = opt(Q, m)$. Тоді S^* розбиває Q за найближчим відношенням знімка. Префікс S^* також є оптимальним розташуванням знімка префікса розбиття Q. Це призводить до рекурсивного вирішення

проблеми розміщення багатьох моментальних знімків, як показано на рис. 2.2.

Base case:

$\text{opt}(Q, 1) = \{\text{median}(Q)\}$

Induction:

$\text{let } i^* = \underset{i \in [1,n]}{\text{argmin}}\ \text{cost}(\text{opt}(Q[1 : i], m - 1))$

$\text{let } n = |Q|$

$\text{opt}(Q, m) = \text{opt}(Q[1 : i^*]) \cup \{\text{median}(Q[i^* + 1 : n])$

Рис. 2.2. Рекурсивне оптимальне формування багатьох моментальних знімків БД

З рис. 2.2 видно випливає, що рекурсію можна реалізувати за допомогою динамічного програмування зі складністю $O(mn^2)$, де m – кількість знімків, а n – кількість запитів.

Для виявлення оптимальних вузлів для розміщення моментальних знімків варто скористатися інтелектуальними методами, зокрема кластерного аналізу.

Хоча алгоритм, наведений на рис. 2.2 обчислює оптимальне розміщення знімка за поліноміальний час, його складність залишається все ще занадто високою. Це підтверджується тим, що у випадку, коли навантаження на запит велике (тисячі рядків даних), то кількість моментальних знімків все ще становить сотні. Тому, на основі методу, що використовує медіану часового ряду є оптимальним рішенням для розміщення у вузлах багатьох знімків бази даних, оскільки використовуються медіани підмножин запитів. Це також справедливо для випадку $m = 1$, як зазначено у першій теоремі. Тому даний факт змушує вважати, що кластерна структура в мітках часу робочого навантаження

запиту є сильним визначальним фактором у розміщенні моментального знімка.

У зв'язку з наведеними аргументами, пропонується апроксимувати оптимальні місця знімків середніми центроїдами m-кластеризації $\{t_q : q \in Q\}$, як показано на рис. 2.3.

approx(Q, m):
    $\mathbf{C}$ = k-mean-clustering($\{t_q : q \in Q\}, m$)
    return {mean(C) : $C \in \mathbf{C}$}

Рис. 2.3. Апроксимація місця розташування моментального знімка БД з використанням алгоритму K-Means

Оскільки k-means має складність $O(mn)$, то відповідно його оптимально використовувати при визначенні місця розташування моментальних знімків. Фактично, існує можливість регулювати час виконання, контролюючи кількість ітерацій над часовими мітками запиту. Зменшивши кількість ітерацій, можна ще більше прискорити обчислення розміщення знімка.

2.6. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. Проаналізовано особливості організації технології блокчейн, зокрема щодо конфіденційності даних та алгоритмів опрацювання транзакцій, що дало змогу виявити потенційні шляхи забезпечення достовірності і безпеки даних, а також масштабування і оптимального опрацювання запитів у розподілених реляційних системах зберігання даних.

2. Запропоновано метод імплементації технології блокчейн для організації класичних розподілених систем зберігання даних шляхом додавання до кожної таблиці бази даних кортежу атрибутів: часова мітка,

цифровий підпис попередньої транзакції, цифровий підпис поточної транзакції, публічний ключ користувача та булевого поля щодо операції видалення, що дало змогу забезпечити достовірність та можливість виявлення несанкціонованого внесення змін у дані незалежно від ролі користувача.

3. Запропоновано метод оптимізації виконання запитів до розподілених систем зберігання даних, який заснований на формуванні одного та багатьох моментальних знімків бази даних з оптимальним їх розташуванням за рахунок кластеризації подібних знімків, що дало змогу підвищити у 50 разів продуктивність опрацювання запитів.

РОЗДІЛ 3
АПРОБАЦІЯ МЕТОДІВ ПІДВИЩЕННЯ ДОСТОВІРНОСТІ ТРАНЗАКЦІЙ ТА ОПТИМАЛЬНОСТІ ВИКОНАННЯ ЗАПИТІВ У РОЗПОДІЛЕНИХ СИСТЕМАХ ЗБЕРІГАННЯ ДАНИХ

3.1. Структура та організація взаємодії між блоками в блокчейні

Будь-яку структуру даних, яка використовується для зберігання інформації, можна вважати базою даних. Технологія блокчейн за своєю суттю — це не більше, ніж журнал для зберігання інформації про транзакції. До цього моменту блокчейни можна вважати базами даних.

Дані зберігаються у вигляді підписаних блоків, які зв’язуються один з одним, створюючи ланцюжок незмінних взаємопов’язаних записів даних. В загальному випадку, блоки у блокчейні можна представити як показано на рис. 3.1.

Рис. 3.1. Блоки у блокчейні

Щоб підписати новий блок, вузол повинен знайти підпис SHA-256, який відповідає певним критеріям. Для цього він використовуватиме поле nonce для перебору можливих рішень.

Будь-який новий блок потрібно перевірити більшістю вузлів, що утворюють блокчейн. Після перевірки блоку він додається до всіх вузлів блокчейну. Цей спосіб перевірки нових блоків називається підтвердженням

роботи (PoW) і був дуже поширеним на початку розвитку технології блокчейн.

Нині з’явилися інші методи підтвердження, такі як підтвердження частки (PoS). Якщо будь-яка інформація в даних усередині блоку змінена, підпис стає недійсним. Щоб знову зробити блок дійсним, цей підпис потрібно змінити. Щоб переконатися, що наступні блоки все ще працюють, для кожного з них також потрібно створити новий підпис. Навіть якщо вузол зможе відновити ці підписи, зміни мають бути прийняті більшістю вузлів, на яких розміщено блокчейн. З цих причин блокчейни є незмінними.

Жодна інформація, що міститься в даних блоків, не може бути змінена. Ними також керує набір децентралізованих вузлів, що усуває потребу в центральному органі для контролю всіх транзакцій. Через цю незмінність блокчейни набули популярності в таких галузях, як фінанси та нерухомість.

Завдяки тому, як працюють блокчейни, вони ідеально підходять для зберігання інформації про активи. У блокчейні можна створювати та передавати активи іншій організації. Ці рухи називаються транзакціями.

Блокчейни можуть здатися чудовим рішенням для зберігання інформації, але вони мають свою ціну. Основне обмеження стосується продуктивності, коли справа доходить до запитів до бази даних. Будь-які нові транзакції мають перевірятися всіма вузлами, і це може бути тривалим процесом, залежно від розміру самого блокчейну.

Запитувати дані також може бути складно, а швидкість операцій зчитування далеко не така, як у базі даних. Ось тут і вступають у гру блокчейн-бази даних. Поєднуючи потужність сучасних баз даних із цілісністю блокчейнів, бази даних блокчейну пропонують спосіб безпечного зберігання даних, водночас забезпечуючи прості способи доступу до даних через транзакції.

Сам по собі блокчейн представляє собою сукупність послідовних взаєпов’язаних модулів. Приклад організації блокчейну показано на рис. 3.2.

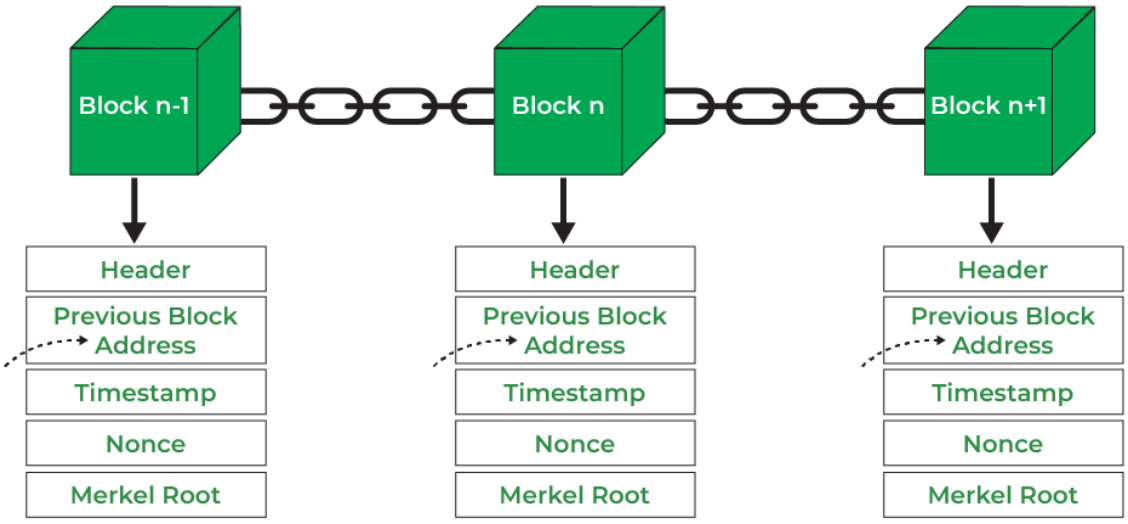

Рис. 3.2. Організація та структура блокчейну

Як видно з рис. 3.2 до складу блоку можуть входити наступні параметри:

– заголовок (Header);
– адреса попереднього блоку (Previous Block Address);
– часова мітка (Timestamp);
– одноразовий номер (Nonce);
– Merkel Root.

Заголовок (Header) використовується для ідентифікації конкретного блоку в усьому блокчейні. Він обробляє всі блоки в блокчейні. Майнери періодично хешують заголовок блоку, змінюючи значення nonce у рамках звичайної діяльності майнінгу. Також у заголовку блоку містяться три набори метаданих блоку.

Попередня адреса блоку/хеш використовується для з’єднання i+1-го блоку з i-м блоком за допомогою хешу. Коротше кажучи, це посилання на хеш попереднього (батьківського) блоку в ланцюжку.

Мітка часу використовується системою для перевірки даних в блоці та призначає час або дату створення для цифрових документів. Мітка часу представляє собою рядок символів, який унікально ідентифікує документ або подію та вказує, коли їх було створено.

Одноразовий номер (Nonce) використовується лише один раз. Це центральна частина підтвердження роботи в блоці. Вона порівнюється з реальною міткою, якщо вона менша або дорівнює поточній мітці. Люди, які видобувають, тестують і усувають багато Nonce за секунду, доки не виявлять, що Valuable Nonce дійсний.

Merkel Root – це тип кадру структури даних різних блоків даних. Дерево Merkle зберігає всі транзакції в блоці, створюючи цифровий відбиток усієї транзакції. Це дозволяє користувачам перевірити, чи можна транзакцію включити до блоку чи ні.

3.2. Типи архітектури блокчейну

Перша з архітектур, які використовуються при побудові блокчейн – загальнодоступна, або по-іншому публічний блокчейн. Дана концепція передбачає, що будь-хто може вільно приєднатися та брати участь в основній діяльності мережі блокчейн.

Будь-хто може читати, писати та перевіряти поточну діяльність у загальнодоступній мережі блокчейн, що допомагає досягти самовизначеного, децентралізованого характеру. Дані в публічному блокчейні захищені, оскільки їх неможливо змінити після перевірки.

Публічний блокчейн повністю децентралізований, він має доступ і контроль над обліковою книгою, його дані не обмежені особами, вони завжди доступні, а центральний орган керує всіма блоками в ланцюжку.

Оскільки ніхто не опрацьовує транзакції окремо, немає необхідності отримувати дозвіл на доступ до загальнодоступного блокчейну. Будь-хто може встановити власний вузол або блок у мережі/ланцюжку. Після того, як вузол або блок імплементований у ланцюжку, усі блоки взаємодіють як однорангові з'єднання. Якщо хтось намагається атакувати блок, він створює копію цих даних і стає доступним лише оригінальному автору блоку.

Приклад організації публічного блокчейну показано на рис. 3.3.

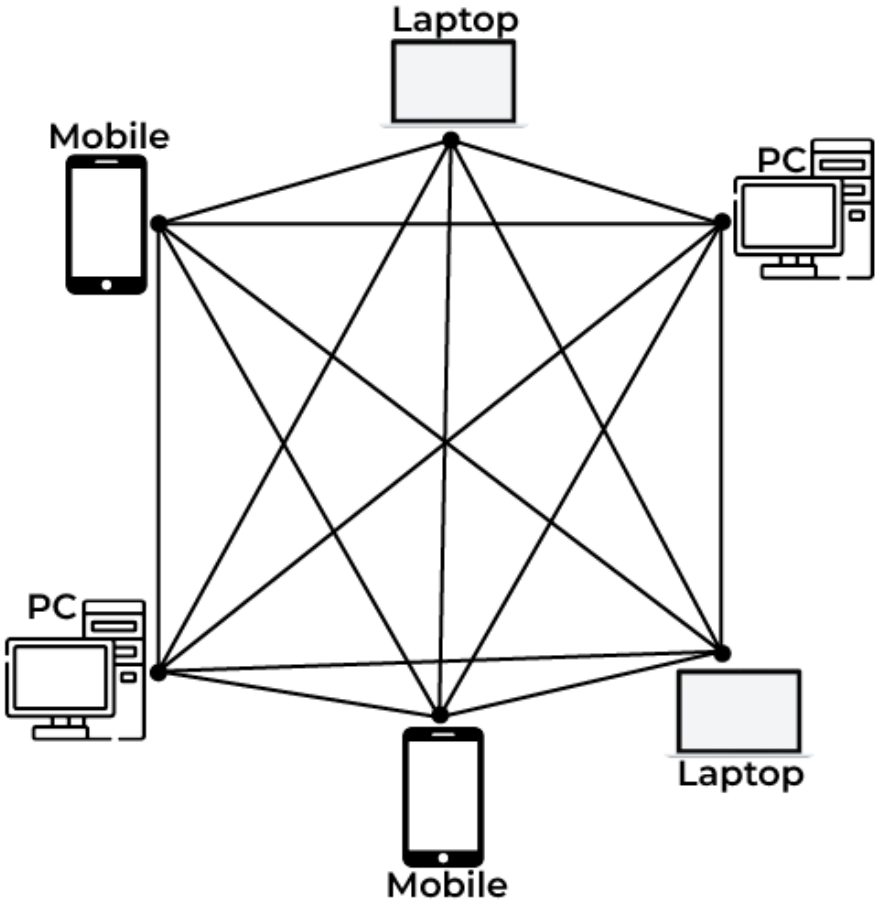

Рис. 3.3. Архітектура публічного блокчейну

До переваг публічного блокчейну належить:

– загальнодоступна мережа працює за схемою активації, яка заохочує нових людей приєднуватися та краще підтримувати мережу.

– відсутність контрактів, що забезпечує незмінність публічної мережі блокчейну.

– швидкість виконання та опрацювання транзакцій.

До недоліків загальнодоступної організації блокчейну можна віднести:

– публічний блокчейн може бути доволі дорогим;

– людині не потрібно вказувати особу, тому існує ймовірність пошкодження блоку, якщо він піддається атаці;

– швидкість опрацювання даних іноді низька;

– існують технологічні проблеми з інтеграцією.

Приватний блокчейн, а відповідно і його архітектурна організація пов’язана з тим, що добувачам криптовалюти потрібен дозвіл на доступ до конкретного блокчейну. Він працює на основі дозволів і елементів керування, які обмежують участь у мережі. Лише суб’єкти, які беруть участь у транзакції, матимуть інформацію про неї, а інші зацікавлені сторони не матимуть.

Така організація блокчейну працює на основі дозволів, тому приватний блокчейн часто називають блокчейном з дозволами.

Приватні блокчейни не схожі на публічні, ними керує організація, яка володіє мережею. Довірена особа відповідає за роботу блокчейну, вона контролює, хто може отримати доступ до приватного блокчейну, а також контролює права доступу до мережі приватного ланцюга. Під час доступу до мережі приватного блокчейну можуть існувати деякі обмеження. На рис. 3.4 показано архітектуру приватного блокчейну.

Перевагами приватного блокчейну є:

– можливість користувачів приєднуватися до мережі за допомогою запрошень, і всі вони перевіряються;

– лише авторизовані користувачі/особи можуть приєднатися до мережі;

– приватний блокчейн є частково незмінним.

До недоліків приватного блокчейну належить:

– приватний блокчейн має проблеми з довірою через те, що до ексклюзивної інформації важко отримати доступ;

– зі збільшенням кількості учасників існує ймовірність атаки на зареєстрованих користувачів.

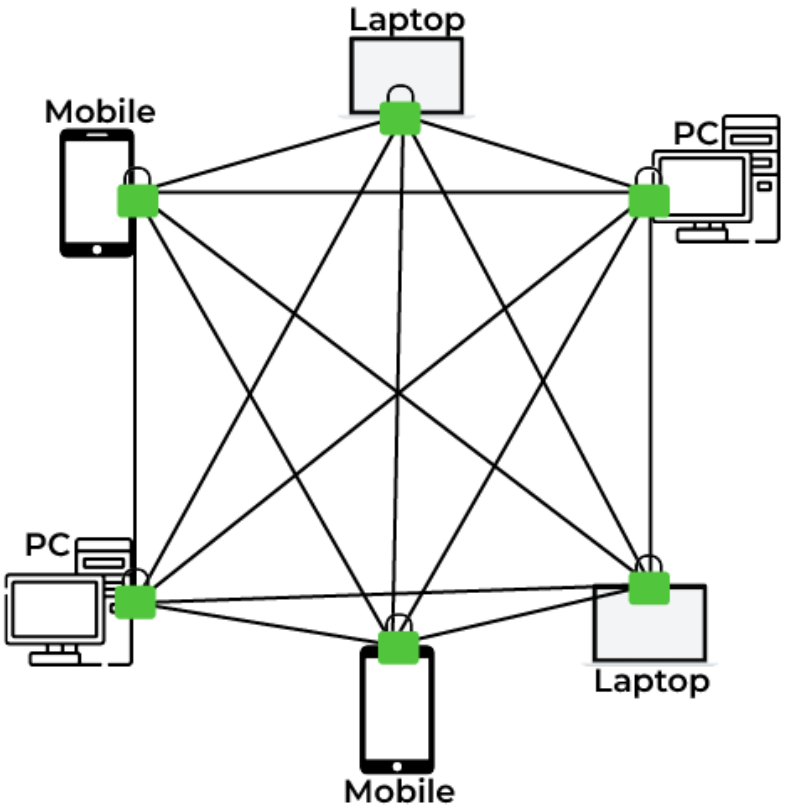

Рис. 3.4. Архітектура приватного блокчейну

Блокчейн консорціуму — це концепція, яка дозволена урядом і групою організацій, а не однією особою, як приватний блокчейн.

Блокчейни консорціуму більш децентралізовані, ніж приватні блокчейни, оскільки більша децентралізація підвищує конфіденційність і безпеку блоків. Приватні блокчейни можуть бути пов’язані з мережею блоків державних організацій.

Блокчейни консорціуму будуються між державними та приватними блокчейнами. Вони розроблені організаціями, і жодна особа поза організаціями не може отримати доступ. У блокчейнах консорціуму всі

компанії між організаціями співпрацюють однаково. Вони не надають доступу ззовні мережі організацій/консорціуму.

Важливими перевагами блокчейнів консорціуму є те, що провайдери блокчейнів завжди намагатимуться надати найшвидший результат порівняно з публічними блокчейнами, він є масштабований з низькими транзакційними витратами.

До недоліків такого типу блокчейнів можна віднести те, що він нестабільний у відносинах, не має економічної моделі та існують проблеми з гнучкістю.

3.3. Відмінності між блокчейном і базою даних

Основні відмінності блокчейн технології та класичних розподілених баз даних показані у вигляді табл. 3.1.

Таблиця 3.1.

Відмінності між блокчейн та РБД

Параметр	Блокчейн	Бази даних
Контроль над даними	Децентралізований	Централізований
Наявність адміністратора	Не потребує, оскільки дані дублюються на кожному вузлі	Потребує збереження та синхронізації даних
Історія змін	У Blockchain наявна інформація в реальному часі, але вона дозволяє відстежувати історію транзакцій.	Традиційні бази даних записують лише поточну інформацію. Таким чином, неможливо відстежити історію транзакцій.

Продовження табл. 3.1

Параметр	Блокчейн	Бази даних
Продуктивність	Блокчейни мають низьку продуктивність у порівнянні з технологією транзакцій.	Реляційні бази даних перевершують блокчейни з точки зору продуктивності.
Цілісність даних	Блокчейн унеможливлює втручання в дані у мережі без розриву ланцюга.	Якщо не вжити необхідних заходів, зловмисник може змінити дані в базах даних.
Операції над даними (транзакції)	У блокчейні можна лише читати або додавати дані до бази даних блокчейну.	База даних дозволяє виконувати операції читання, запису, оновлення та видалення даних.
Дозволи	Блокчейни не мають дозволу, оскільки будь-хто може отримати до них доступ.	Бази даних передбачають наявність дозволів, оскільки доступ до них мають лише особи, які мають відповідні права
Конфіденційність.	Блокчейн безпечний і конфіденційний	Бази даних не є повністю конфіденційними.
Рекурсивність.	Блокчейни не є рекурсивними, оскільки неможливо повернутися до повторного завдання на будь-якому записі	Бази даних є рекурсивними, оскільки можна повернутися до певного запису, щоб повторити завдання.

Таким чином, існують значені відмінності між блокчейном та типовими реляційними базами даних. Однак для підвищення надійності та захисту даних потрібно розробити процедури, які б на технічному рівні давали змогу використовувати переваги одних та інших підходів до опрацювання та зберігання даних.

3.4. Реалізація каркасу блокчейну мовою програмування Python

Реалізація розподіленої бази даних на основі блокчейну доволі складна і не тривіальна задача. Проте організація каркасу блокчейну мовою програмування Python дає змогу реалізувати основні її концепції досить просто. Для цього пропонується скористатися бібліотекою pycrypto. Демонстрація реалізації базових компонентів блокчейну, як розподліеної бази даних показано на рис. 3.5 і рис. 3.6.

```
import hashlib
import time
class Block:
    def __init__(self, index, previous_hash, data, timestamp=None):
        self.index = index
        self.previous_hash = previous_hash
        self.data = data
        self.timestamp = timestamp or time.time()
        self.hash = self.calculate_hash()
    def calculate_hash(self):
        data_string = f"{self.index}{self.previous_hash}{self.timestamp}{str(self.data)}"
        return hashlib.sha256(data_string.encode()).hexdigest()
class Blockchain:
    def __init__(self):
        self.chain = [self.create_genesis_block()]

    def create_genesis_block(self):
        return Block(0, "0", "Genesis Block", time.time())

    def get_latest_block(self):
        return self.chain[-1]

    def add_block(self, new_block):
        new_block.previous_hash = self.get_latest_block().hash
        new_block.hash = new_block.calculate_hash()
        self.chain.append(new_block)
```

Рис. 3.5. Створення класів «Block» та «Blockchain»

```python
# Створимо блокчейн та додамо кілька блоків
my_blockchain = Blockchain()
my_blockchain.add_block(Block(1, my_blockchain.get_latest_block().hash, {"amount": 10, "sender": "Alice", "receiver": "Bob"}))
my_blockchain.add_block(Block(2, my_blockchain.get_latest_block().hash, {"amount": 5, "sender": "Bob", "receiver": "Charlie"}))

# Виведемо ланцюжок блоків
for block in my_blockchain.chain:
    print(f"Block #{block.index}")
    print(f"Timestamp: {block.timestamp}")
    print(f"Data: {block.data}")
    print(f"Previous Hash: {block.previous_hash}")
    print(f"Hash: {block.hash}")
    print()
```

Рис. 3.6. Програмний код створення і відображення блокчейну

У цьому прикладі створено простий клас Block, який представляє окремий блок у блокчейні, і клас Blockchain, який управляє ланцюжком блоків.

У лістингу (рис. 3.6) створено генезис-блок як початок ланцюжка та додано кілька інших блоків, які посилаються на попередній блок та містять деяку інформацію у вигляді словників.

Слід зауважити, що цей код є дуже спрощеним і не враховує багато важливих аспектів, які вимагаються для реалізації реального блокчейну, такі як консенсусні алгоритми, мережева комунікація, зберігання та безпека даних.

Реалізація розподіленої бази даних на основі блокчейну є складною задачею, і для реального застосування потребує додаткового дослідження.

3.5. Практична реалізація та експериментальне застосування методу організації розподілених систем зберігання даних

Для проведення експериментів щодо застосування запропонованого методу організації розподілених комп’ютерних систем із застосуванням принципів технології блокчейн необхідно спочатку створити сховище із застосуванням реляційного підходу до організації баз даних. В якості прикладу створимо елементарну базу даних без імплементації шифрування даних, часових міток і т.п., яка орієнтована на зберігання та опрацювання

даних електронної комерції. Середовище, що використовується для формування бази даних – MS SQL Server.

Структура бази даних міститиме таблиці, які призначенні для опису наступних сутностей:

– товар;
– категорія товару;
– продавець;
– покупець;
– замовлення;
– стан замовлення;
– транзакція;

SQL-запит на формування таблиць товарів і категорії товарів показано на рис. 3.7.

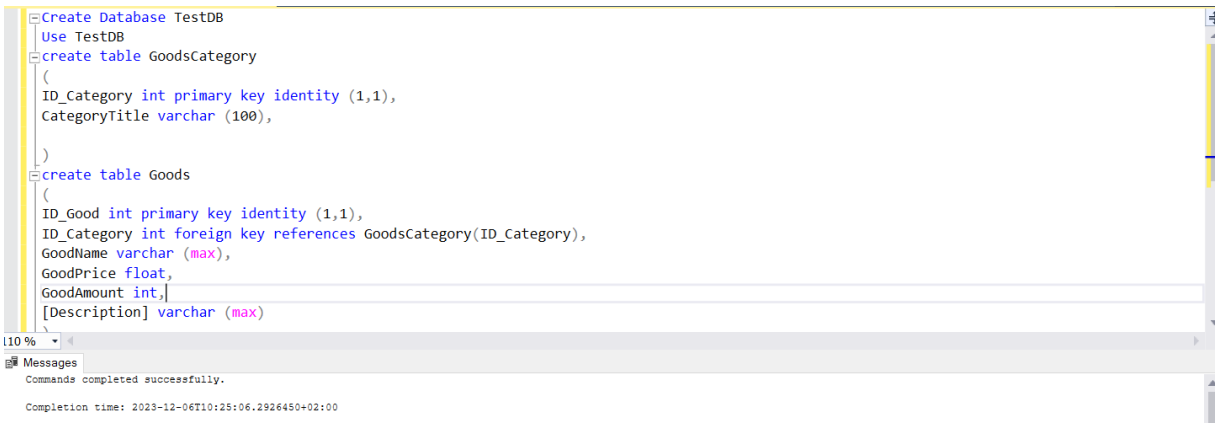

Рис. 3.7. SQL-запит для фізичної реалізації сутностей товарів та категорії товарів

При фізичній реалізації сутності «Категорія товарів» оголошено лише атрибути:

– ідентифікатор категорії товарів;
– назва категорії.

Таблиця товарів включає атрибути, які описують їх властивості місять:

– ідентифікатор товару;
– назва товару;
– ціна;
– кількість;
– опис.

Варто відмітити, що зв’язок між категорією товару та самим товаром забезпечується через використання зовнішнього ключа. При цьому інтерпретується він таким чином: «До однієї категорії може належати багато товарів». По аналогії створено інші таблиці бази даних, структуру якої представлено на рис. 3.8.

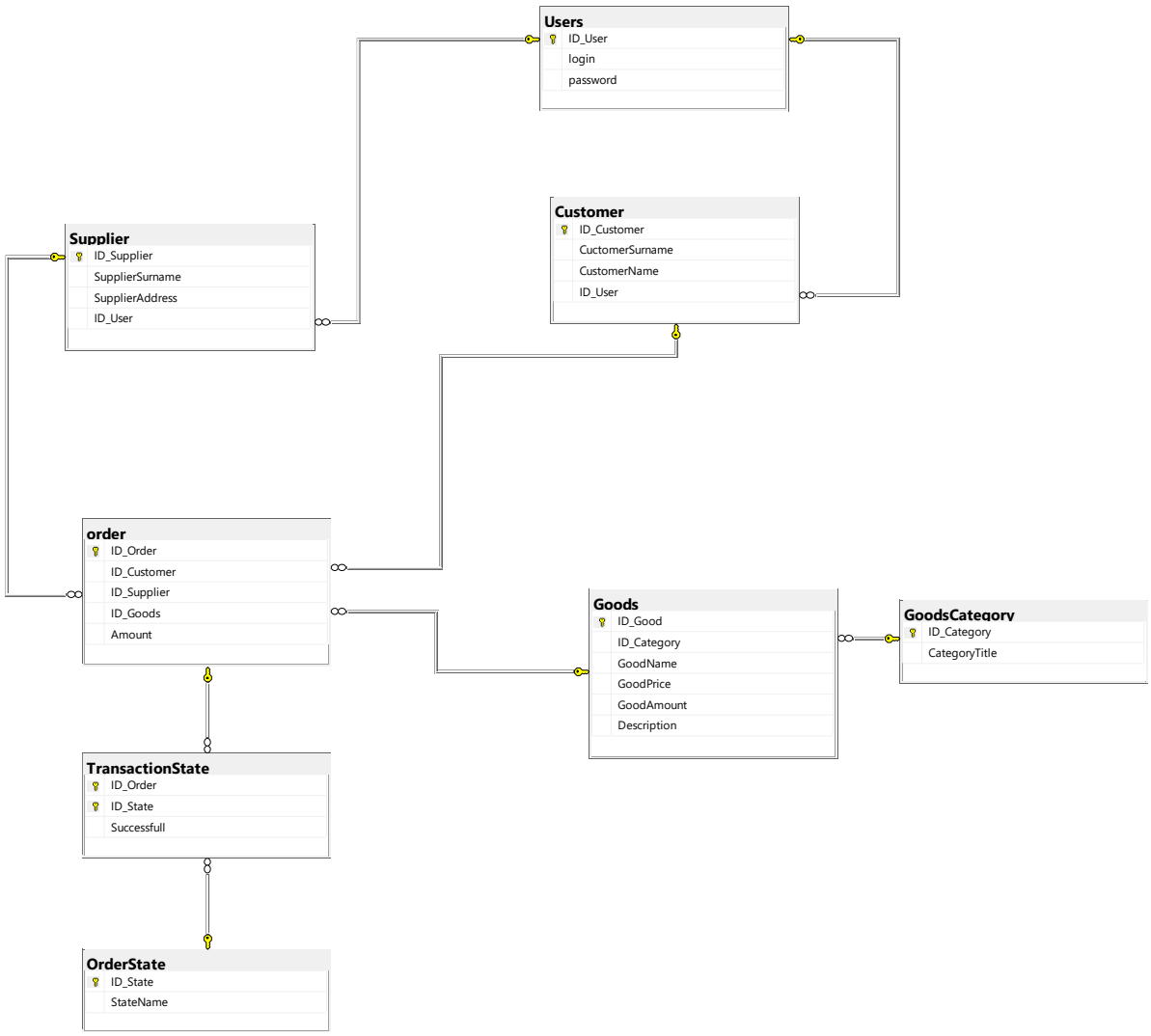

Рис. 3.8. Діаграма «сутність-зв’язок» тестової бази даних

У результаті проведених маніпуляцій та реконструкції бази даних одержано схему, яка показана на рис. 3.9.

Рис. 3.9. Схема бази даних з інтеграцією технології блокчейн

У запропонованому методі, який використовує принцип технології блокчейн, необхідно для кожної таблиці додати визначені у розділі 2 атрибути:

– TimeLabel – часова мітка;

– PreviousUserSign – цифровий підпис користувача, який провів попередню транзакцію,

– CurrentUserSign – цифровий підпис користувача, який провів поточну транзакцію,

– PublicKey – публічний ключ користувача (можливе використання ідентифікатора користувача);

– Del – булеве поле для визначення транзакції на видалення даних.

Як видно з рис. 3.8 та рис. 3.9 схема організації бази даних з імплементованими елементами технології блокчейн відрізняється лише доданими до структури таблиць додаткових полів. Це означає, що цілісність бази даних є непорушною. Далі структуру БД можна розподілити за вузлами мережі з врахуванням факторів оптимальності використання даних, при цьому найбільш ефективним є застосування горизонтальної фрагментації.

У роботі було проведено ряд експериментів, щоб оцінити ефективність запропонованих алгоритмів. Робоче навантаження синтетичного запиту створюється для бази даних із понад 1 мільйоном кортежів. На рис. 3.10 – 3.12 показано переваги оптимального розміщення моментальних знімків. Рис. 3.10 відображає ефект опрацювання запиту за наявності лише одного знімка.

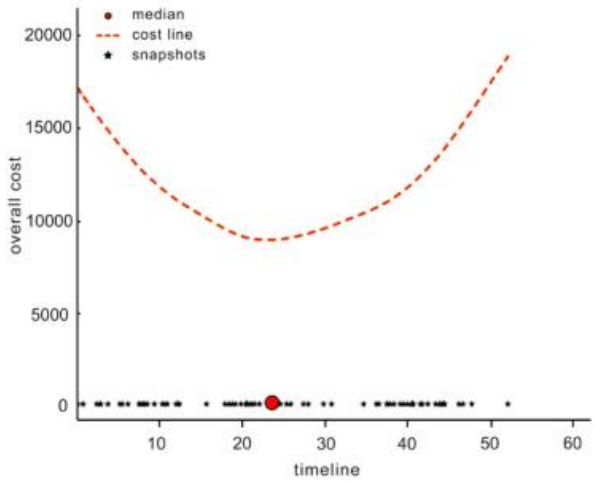

Рис. 3.10. Оптимальність виконання запитів при застосуванні одного моментального знімка бази даних

З рис. 3.10 можна побачити, що вартість виконання розподіленого запиту мінімальна, коли моментальний знімок знаходиться на рівні медіани робочого навантаження запиту.

На рис. 3.11 показано переваги алгоритму оптимального виконання запиту до розподіленої системи зберігання даних за наявності більшої кількості оптимально розподілених моментальних знімків.

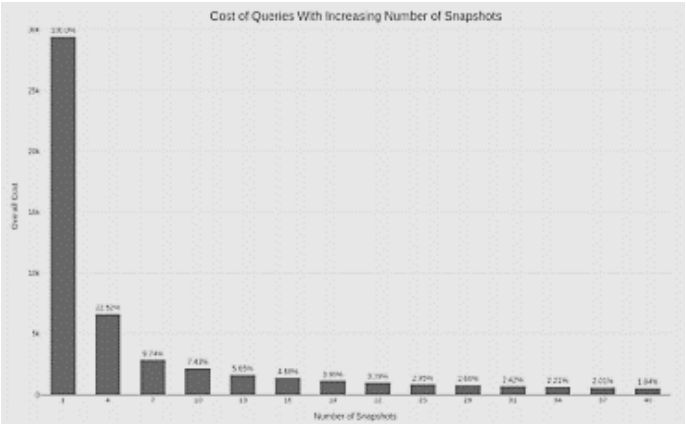

Рис. 3.11. Оптимальність виконання запитів за наявності багатьох моментальних знімків БД

З рис. 3.11 видно, що з 40 моментальними знімками вартість зменшується до менш ніж 2% порівняно з лише одним знімком. На рис. 3.12 проілюстровано відносну ефективність різних стратегій розміщення знімків.

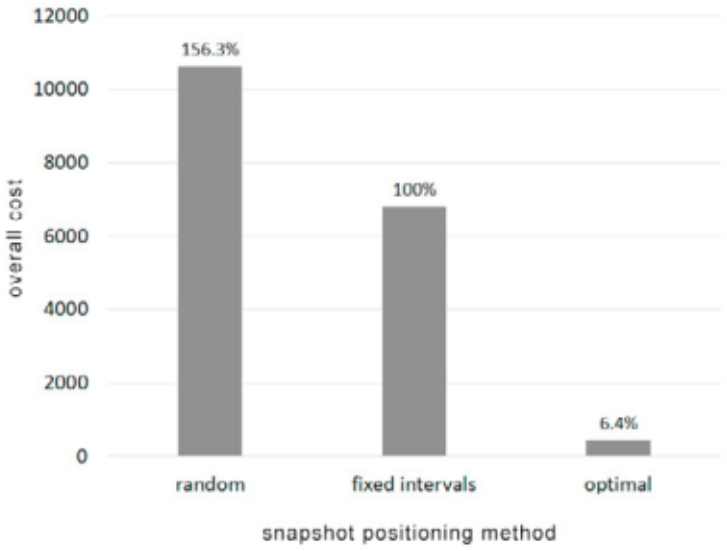

Рис. 3.12. Відносна ефективність при різних стратегіях розміщення знімків

Випадкові розміщення та розміщення з фіксованим інтервалом є значно гіршими за оптимальну стратегію розміщення, а вартість у 15 разів гірша. Також проведено порівняння продуктивності під час виконання оптимального розміщення знімка за допомогою динамічного програмування (рекурсії) та евристики на основі кластеризації.

На рис. 3.13 порівнюється продуктивність виконання динамічного програмування та кластеризації з 300 ітераціями.

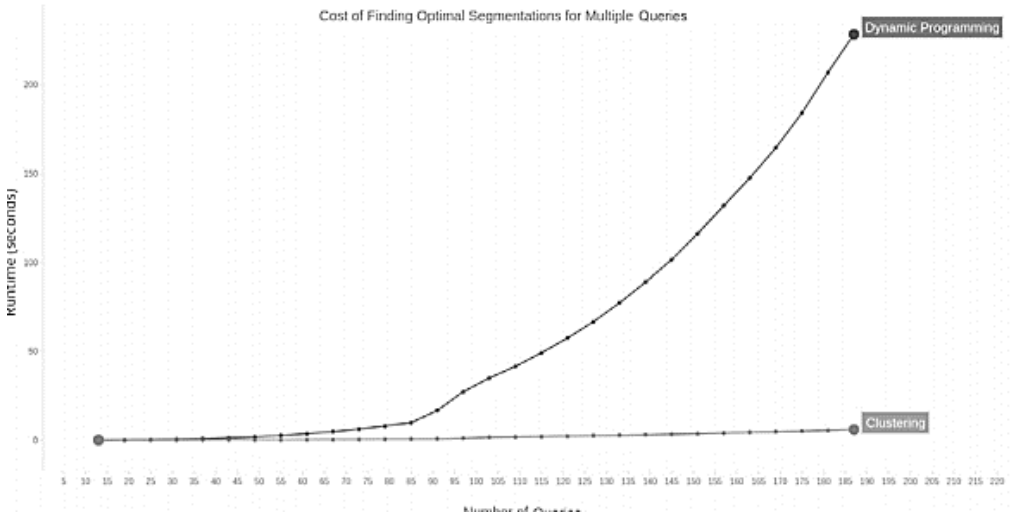

Рис. 3.13. Результати порівняння продуктивності із застосуванням кластеризації та рекурсії

Як видно з рис. 3.13 евристика кластеризації надзвичайно добре масштабується зі збільшенням розміру робочого навантаження запиту. Щоб ще більше прискорити обчислення розміщення знімків, можна зменшити кількість ітерацій, як продемонстровано на рис.3.14.

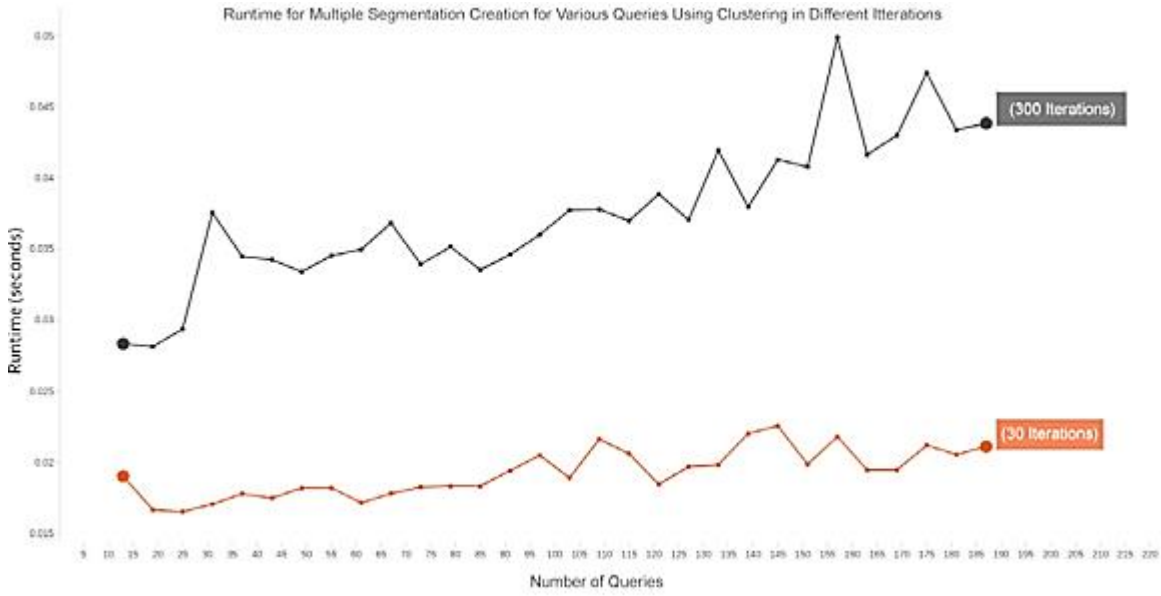

Рис. 3.14. Порівняння продуктивності із зменшеною кількістю ітерацій

Евристичний підхід є дуже ефективним для отримання майже оптимального розташування знімків. На рис. 3.15 порівнюється витрати оптимальних розміщень і приблизних розміщень з використанням 300 ітерацій.

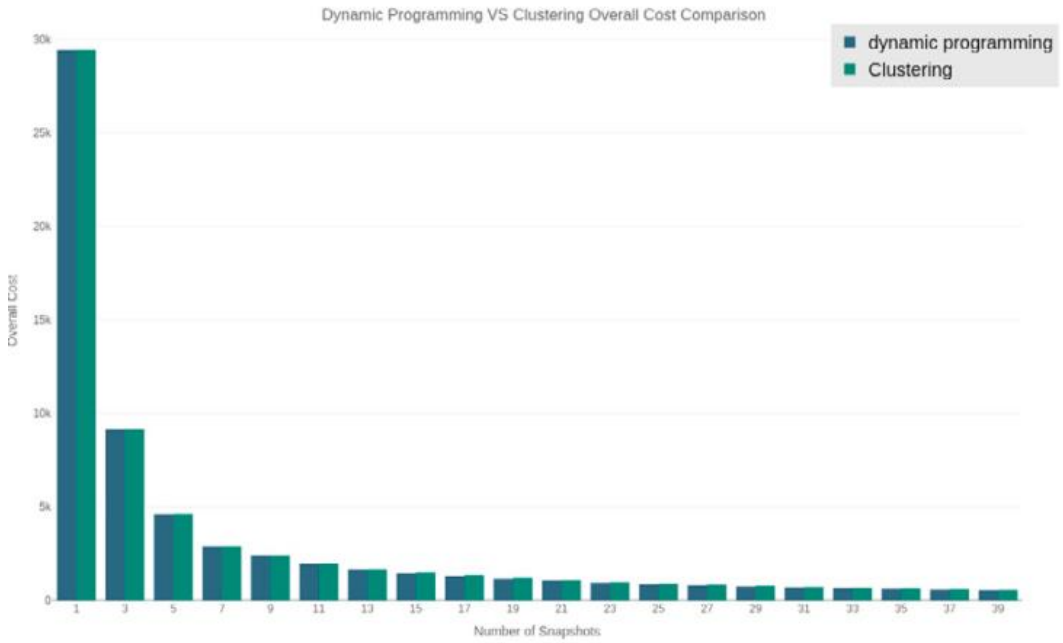

Рис. 3.15. Порівняння витрат з використанням оптимальних і приблизних розміщень моментальних знімків

Як видно з результатів, наведених на рис. 3.15, спостерігається збільшення вартості менше ніж 2%. Навіть із лише 5 ітераціями, як проілюстровано на рис. 3.16, отримане наближення дуже близьке до точного оптимального розміщення

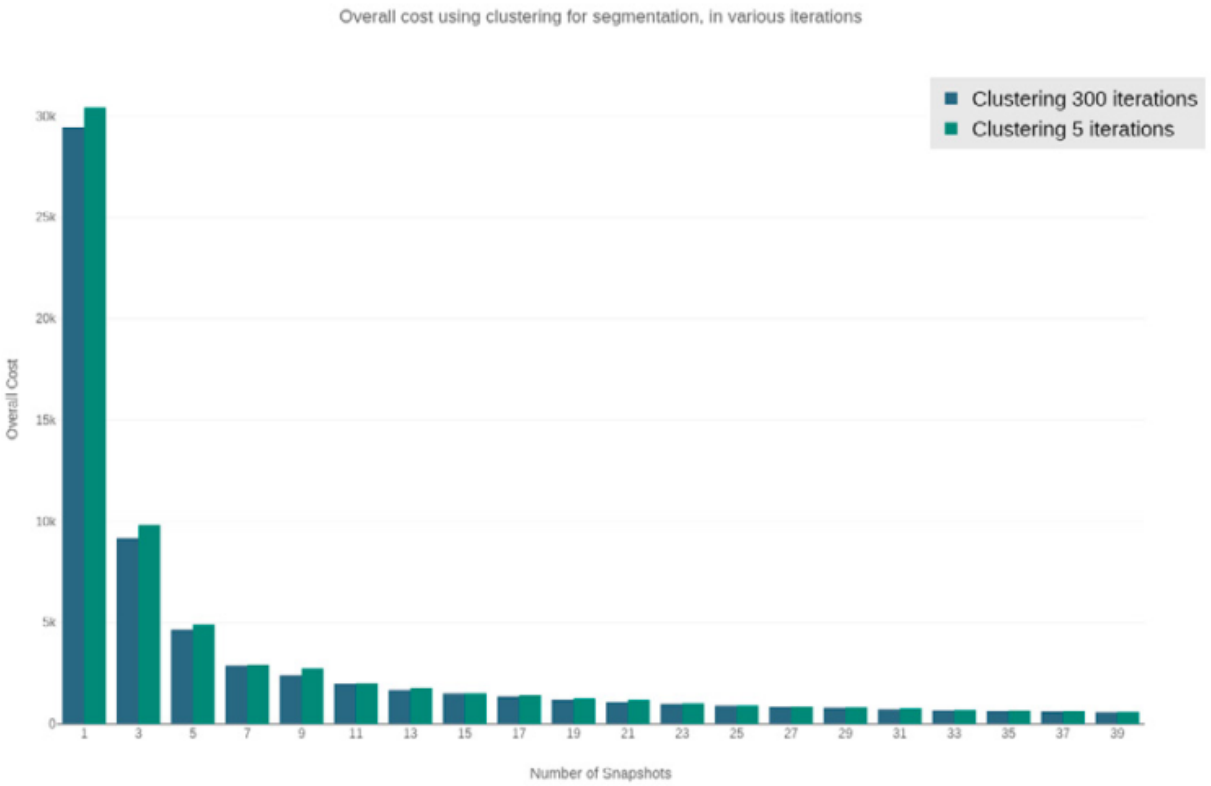

Рис. 3.16. Кластеризація за 5 і 300 ітераціями

Таким чином, експериментально обгрунтовано доцільно застосування методу організації розподілених систем зберігання даних з використанням елементів підходу блокчейн та реляційних баз даних.

Такі система підтримують транзакційне опрацювання даних з великою кількістю користувачів і пристроїв, а також можливість верифікації за допомогою вбудованих блокчейнів, які зберігають пов’язану з довірою інформацію щодо усіх транзакцій. База даних підтримує аналітичні запити з різними часовими мітками. Щоб досягти високої ефективності, запропонуємо оптимально розміщати моментальні знімки з різними часовими мітками, щоб мінімізувати витрати часу на оцінку запиту.

3.6. Висновки до розділу

Основні результати даного розділу полягають в наступному:

1. Визначено особливості організації структури даних у блокчейнах і принципів функціонування транзакційного механізму, що дало можливість встановити потенційні шляхи імплементації технології блокчейн у класичних розподілених системах зберігання даних.

2. Визначено ключові відмінності між блокчейном та розподіленими базами даних, що дало змогу поєднати переваги обох способів організації розподілених систем зберігання даних та імплементувати механізми забезпечення достовірності транзакцій, підвищення рівня їх захищеності і оптимальності розподілу моментальних знімків бази даних за вузлами мережі.

3. Реалізовано каркас блокчейну засобами мови програмування Python для подальшого його трансформування на реляційні структури систем зберігання даних.

4. Проведено експериментальні дослідження щодо практичної реорганізації розподілених реляційних баз даних з використанням оптимізації на основі кластерного аналізу моментальних знімків БД, що дало змогу значно підвищити (у 50 разів) пропускну здатність опрацювання запитів та забезпечити оптимальність розміщення моментальних знімків (з відхиленням до 2%) за вузлами мережі у порівнянні з точними оптимальними розміщеннями.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Тема кваліфікаційної роботи магістра пов’язана із дослідженням технологій створення розподілених комп’ютерних систем зберігання даних на основі блокчейн. Такі роботи передбачають використання комп’ютерної техніки на етапах формування пояснювальної записки і налаштування засобів автоматизації при створенні розподілених систем зберігання даних з використанням блокчейн. Тому, при виконанні таких робіт необхідно враховувати вимоги з охорони праці при експлуатації комп’ютерної техніки.

В Україні діють ряд законів, нормативних документів та актів, які регулюють процеси забезпечення та управління охороною праці у різних галузях народного господарства. До них належать: Конституція України, Закони України "Про охорону праці", "Про охорону здоров'я", "Про пожежну безпеку", "Про забезпечення санітарного та епідемічного благополуччя населення", "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності", Кодекс законів про працю України (КЗпП).

Однією з основних вимог до приміщень, де робочі місця обладнані комп’ютерною технікою і планується використання засобів для роботи з розподіленими комп’ютерними системами зберігання даних, є вимоги щодо площі, яка відводиться на один ПК. При проектуванні автоматизованих робочих місць адміністраторів систем зберігання даних необхідно дотримуватись вимог щодо розміщення комп’ютерів. На один ПК передбачено площу 6 м^2 та об’єм 20 м^3.

Однак робота з комп’ютером включає різні завдання, які об'єднуються

такими загальними чинниками, як те, що робота проводиться в сидячому положенні і вимагає уважного, неперервного та іноді тривалого спостереження.

Перше правило, якого варто дотримуватись адміністраторам розподілених комп’ютерних систем зберігання даних на основі блокчейн стосується правильного облаштування робочого столу. При цьому слід передбачити наступні його параметри: фіксована висота – 720 мм, забезпечення необхідного простору для рук по висоті, ширині і глибині, в області сидіння не повинно бути шухляд.

Друге правило визначає облаштування робочого стільця: можливість регулювання висоти стільця, забезпечення обертання конструкції стільця. У приміщеннях з ПК, на яких планується виконання задач з адміністрування розподілених комп’ютерних систем зберігання даних, яскравість знаків і яскравість фону дисплею повинна бути спроектована таким чином, щоб не було великої відмінності з яскравістю навколишнього середовища, але знаки повинні чітко розпізнаватися на відстані читання. Характеристики освітлення, зокрема у приміщеннях, де експлуатується ПК, повинні відповідати ДБН В.2.5-28-2018 "Природнє і штучне освітлення". Основні вимоги даного нормативного документу стосуються забезпечення наступних вимог:

– освітлення з лівої сторони;

– рівномірне освітлення всього робочого простору;

– комп’ютерна техніка встановлюється у місцях, віддалених від вікон;

– встановлення непрямого штучного освітлення;

– світло, що поступає через вікна, «пом'якшують» за допомогою штор;

– робоче місце організовується так, щоб напрям погляду був паралельним фронту вікон.

Ще одне правило, якого слід дотримуватись адміністраторам

розподілених комп’ютерних систем зберігання даних на основі блокчейн, передбачає оптимальний метод роботи, що полягає у передбачені зміни завдань і навантажень, дотримання перерви в роботі: 5 хвилин через 1 годину роботи біля дисплея або 10 хвилин після 2-х годин роботи біля дисплея. Вимоги цього правила регламентовані нормативним документом «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

При створенні сприятливих умов для підвищення продуктивності і зменшення напруги значну роль грають чинники, що характеризують стан навколишнього середовища: мікроклімат приміщення, рівень шуму і освітлення.

Рекомендована величина відносної вологості, яка повинна бути забезпечена у приміщеннях з експлуатації комп’ютерних систем, повинна відповідати НПАОП 0.00-7.15-18 і становити 65 – 70%. При цьому робоче місце повинно бути добре вентильованим.

У даний час з погляду шумового навантаження досягнуто значного прогресу. Рівень шуму в приміщеннях (приблизно 40 Дб) не перевищує допустимого рівня, незалежно від кількості використовуваного обладнання. Для приміщень, в яких використовуються технології розробки та управління розподіленими комп’ютерними системами зберігання даних, потрібно забезпечити виконання вимог пожежної безпеки, які визначені Правилами пожежної безпеки в Україні, НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [19].

Будівлі і ті їх частини, в яких розташовуються ПК можуть належати до II ступеня вогнестійкості. Над та під приміщеннями, де розташовуються ПК, а також у суміжних з ними приміщеннях не дозволяється розташування приміщень категорій А і Б за вибухопожежною небезпекою. Приміщення категорії В повинні бути відділеними від приміщень з ПК протипожежними стінами.

Таким, чином при дослідженні технологій створення розподілених комп’ютерних систем зберігання даних на основі блокчейн, встановлено, що найбільш повним нормативним документом щодо охорони праці користувачів ПК, до яких належать тестувальники програмного забезпечення, є НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Дотримання вимог, які неведені у цьому документі, сприяє зниженню негативного впливу ПК, його компонентів та інших зовнішніх пристроїв на адміністраторів розподілених комп’ютерних систем зберігання даних, які проводять роботи щодо підтримки їх цілісності та коректного розгортання і налаштування.

4.2 Освітлення виробничих приміщень для роботи з ВДТ та локальній комп’ютерній мережі

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до ДБН В.2.5-28:2018 (на заміну ДБН В.2.5-28-2006).

Природне освітлення повинно потрапляти через світлові прорізи, орієнтовані переважно на північ чи північний схід і забезпечувати коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%.

Розраховується КПО за методикою, викладеною в ДБН В.2.5-28:2018. За виробничої потреби дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами державного нагляду за охороною праці та органами і установами санітарноепідеміологічної служби. Вікна приміщень з ВДТ повинні мати регулювальні пристрої для відкривання, а також жалюзі, штори, зовнішні козирки тощо.

Штучне освітлення приміщення з робочими місцями, обладнаними ВДТ ЕОМ загального та персонального користування, має бути обладнане системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, де переважають роботи з

документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення).

Загальне освітлення має бути виконане у вигляді суцільних або переривчатих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. Допускається застосовувати світильники таких класів світлорозподілу:

- світильники прямого світла – П;
- переважно прямого світла – Н;
- переважно відбитого світла – В.

При розташуванні відеотерміналів ЕОМ за периметром приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями. Для загального освітлення необхідно застосовувати світильники із розсіювачами та дзеркальними екранними сітками або віддзеркалювачами.

Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи типу ЛБ. При обладнанні відбивного освітлення у виробничих та адміністративно-громадських приміщеннях можуть застосовуватися металогалогенові лампи потужністю до 250 Вт.

Допускається у світильниках місцевого освітлення застосовувати лампи розжарювання. Яскравість світильників загального освітлення в зоні кутів випромінювання від 50о до 90о відносно вертикалі в подовжній і поперечній площинах повинна складати не більше 200 кд/м^2 , а захисний кут світильників повинен бути не більшим за 40°.

Коефіцієнт запасу (Кз) для освітлювальної установки загального освітлення слід приймати рівним 1,4. Коефіцієнт пульсації повинен не перевищувати 5% і забезпечуватися застосуванням газорозрядних ламп у світильниках загального і місцевого освітлення. При відсутності світильників з ВЧ ПРА лампи багатолампових світильників або розташовані поруч світильники загального освітлення необхідно підключати до різних

фаз трифазної мережі. Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300–500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж 300 лк. Світильники місцевого освітлення повинні мати напівпрозорий відбивач світла з захисним кутом не меншим за 40°.

Варто передбачити обмеження прямого відбиття від джерела природного та штучного освітлення, при цьому яскравість поверхонь, що світяться (вікна, джерела штучного світла) і перебувають у полі зору, повинна бути не більшою за 200 кд/м^2.

Окрім цього, додвільно обмежувати відбитий блиск шляхом правильного вибору типів світильників та розміщенням робочих місць відносно джерел природного та штучного освітлення. При цьому яскравість відблисків на екрані відеотерміналу не повинна перевищувати 40 кд/м^2, яскравість стелі при застосуванні системи відбивного освітлення не повинна перевищувати 200 кд/м^2.

Серед іншого, варто обмежувати нерівномірність розподілу яскравості в полі зору осіб, що працюють з відеотерміналом, при цьому відношення значень яскравості робочих поверхонь не повинно перевищувати 3:1, а робочих поверхонь і навколишніх предметів (стіни, обладнання) – 5:1.

Необхідно використовувати систему вимикачів, що дозволяє регулювати інтенсивність штучного освітлення залежно від інтенсивності природного, а також дозволяє освітлювати тільки потрібні для роботи зони приміщення. Необхідно очищати віконне скло та світильники не рідше ніж 2 рази на рік, та своєчасно проводити заміну ламп, що перегоріли.

4.3 Дослідження стійкості роботи суб’єкта господарювання у надзвичайних ситуаціях та його програмне забезпечення автоматизованої комп’ютерної обробки

Значні руйнування, пожежі та втрати серед населення, викликані наслідками НС, можуть стати причиною різкого скорочення випуску промислової та сільськогосподарської продукції, а отже і зниження економічного потенціалу держави. Виникає потреба завчасного вживання заходів щодо забезпечення стійкої роботи промислових об’єктів на випадок виникнення НС.

Знання можливих НС, характерних для даної місцевості та виробництва, дозволяє диференційовано і цілеспрямовано розробляти та здійснювати заходи, які можуть запобігти аваріям, катастрофам та стихійним лихам або пом’якшити їх наслідки [20].

Стійкість роботи об’єкта господарської діяльності – це здатність його в умовах НС випускати продукцію у запланованому обсязі та визначеної номенклатури, а у разі слабких та середніх руйнувань або порушення матеріального постачання - відновлювати виробництво власними силами у короткий термін.

На стійкість роботи об’єкта впливають такі фактори:

– захищеність робітників та службовців від уражальних факторів у НС;

– здатність інженерно-технічного комплексу об’єкта (будівель, споруд, обладнання та комунально-енергетичних мереж) протистояти руйнівній дії уражальних факторів аварій, катастроф, стихійного лиха та сучасної зброї;

– надійність постачання об’єкта електроенергією, водою, паливом, комплектуючими та сировиною;

– підготовленість об’єкта до проведення аварійно-рятувальних та відновлюваних робіт;

– оперативність управління виробництвом та здійсненням заходів ЦЗ у НС.

Підвищення стійкості об’єкта досягають проведенням комплексу інженернотехнічних, технологічних, організаційних заходів. До інженерно-технічних заходів належать роботи, що забезпечують стійкість виробничих будівель і споруд, обладнання та комунально-енергетичних систем.

Технологічні заходи забезпечують підвищення стійкості об’єкта спрощенням технологічного процесу виробництва кінцевої продукції та виключенням або обмеженням розвитку аварій.

Організаційні заходи передбачають розробку ефективних дій керівного складу, служб та формувань ЦЗ, спрямованих на захист виробничого персоналу, проведення рятувальних та інших невідкладних робіт, а також відновлення виробництва [21].

Для забезпечення надійного управління діяльністю об'єкта у надзвичайних ситуаціях воєнного часу в одному із сховищ обладнується пункт управління. Диспетчерські пункти і радіовузли розміщують по можливості у найміцніших спорудах і підвальних приміщеннях. Повітряні лінії зв'язку до найважливіших виробничих ділянок переводять на підземно-кабельні [20].

Стійкість засобів зв'язку можна підвищити прокладанням підземно-кабельних ліній на автоматичну телефонну станцію (АТС) та радіовузол об'єкта, підготовкою пересувних електростанцій для заряджання акумуляторів і для живлення радіовузла при відключенні основних джерел електропостачання.

При розширенні мережі підземних кабельних ліній необхідно прокладати дводротові, захищені екранами від впливу ЕМІ (електромагнітних імпульсів). Для більшої надійності повинні бути передбачені дублюючі засоби зв'язку.

У районі розосередження робітників і службовців також обладнують пункт управління. Між міським і заміським пунктами управління проводять

зв'язок, як правило, телефонний, передбачаючи його дублювання за допомогою радіо- та пересувних засобів, також вживають заходів по забезпеченню зв'язку із змінними підприємствами по кооперації.

Особливе значення має сталість виробничих та господарських зв'язків з постачання об'єкта всіма видами енергії, водою, парою, газом; з транспортних послуг; з поставок сировини, напівфабрикатів, комплектуючих виробів та ін. Підвищення сталості матеріально-технічного постачання забезпечується створенням запасів сировини, матеріалів, комплектуючих виробів, обладнання, палива. Розміри незменшуваних запасів визначають для кожного об'єкта залежно від можливості їх накопичення, важливості продукції, яка випускається, визначених термінів переходу на виробництво продукції в умовах надзвичайних ситуацій.

Стабільно працююче підприємство повинно бути здатним безперебійно випускати продукцію за рахунок наявних запасів до відновлення зв'язків з поставок або до одержання необхідного від нових постачальників.

Висновки.

Дотримання норм і вимог щодо облаштування природного та штучного освітлення згідно з ДБН В.2.5-28:2018 забезпечує оптимальні умови праці користувачів комп’ютерів та сприяє зменшенню навантаження на зоровий аналізатор людини.

Заходи щодо забезпечення стійкості роботи суб’єкта господарювання у надзвичайних ситуаціях та відповідного програмного забезпечення комп’ютерної системи автоматизації полягає у формуванні резервних каналів зв’язку та розміщення центрів управління, які його використовують у спеціалізованих сховищах для захисту апаратного забезпечення від наслідків надзвичайних ситуацій.

ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Проведено аналіз основних понять, якими оперують при організації розподілених систем зберігання даних, визначено переваги і недоліки застосування таких систем, що дало змогу визначити шляхи оптимізації при формуванні та виконанні транзакцій з великим об’ємом даних.

2. Проаналізовано класи розподілених систем зберігання даних та визначено, що основними з них є гомогенні та гетерогенні системи, які відрізняються як на рівні типів використовуваного апаратного, так і програмного забезпечення.

3. Проведено аналітичний огляд архітектур розподілених систем зберігання даних та визначено рівні їх організації, зокрема концептуальний, зовнішній та внутрішній, які дали змогу оцінити можливість оптимізації в контексті виконання розподілених транзакцій.

4. Проаналізовано особливості організації технології блокчейн, зокрема щодо конфіденційності даних та алгоритмів опрацювання транзакцій, що дало змогу виявити потенційні шляхи забезпечення достовірності і безпеки даних, а також масштабування і оптимального опрацювання запитів у розподілених реляційних системах зберігання даних.

5. Запропоновано метод імплементації технології блокчейн для організації класичних розподілених систем зберігання даних шляхом додавання до кожної таблиці бази даних кортежу атрибутів: часова мітка, цифровий підпис попередньої транзакції, цифровий підпис поточної транзакції, публічний ключ користувача та булевого поля щодо операції видалення, що дало змогу забезпечити достовірність та можливість виявлення несанкціонованого внесення змін у дані незалежно від ролі користувача.

6. Запропоновано метод оптимізації виконання запитів до розподілених систем зберігання даних, який заснований на формуванні одного та багатьох моментальних знімків бази даних з оптимальним їх розташуванням за рахунок кластеризації подібних знімків, що дало змогу підвищити у 50 разів продуктивність опрацювання запитів.

7. Визначено особливості організації структури даних у блокчейнах і принципів функціонування транзакційного механізму, що дало можливість встановити потенційні шляхи імплементації технології блокчейн у класичних розподілених системах зберігання даних.

8. Визначено ключові відмінності між блокчейном та розподіленими базами даних, що дало змогу поєднати переваги обох способів організації розподілених систем зберігання даних та імплементувати механізми забезпечення достовірності транзакцій, підвищення рівня їх захищеності і оптимальності розподілу моментальних знімків бази даних за вузлами мережі.

9. Реалізовано каркас блокчейну засобами мови програмування Python для подальшого його трансформування на реляційні структури систем зберігання даних.

10. Проведено експериментальні дослідження щодо практичної реорганізації розподілених реляційних баз даних з використанням оптимізації на основі кластерного аналізу моментальних знімків БД, що дало змогу значно підвищити (у 50 разів) пропускну здатність опрацювання запитів та забезпечити оптимальність розміщення моментальних знімків (з відхиленням до 2%) за вузлами мережі у порівнянні з точними оптимальними розміщеннями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ярцев В.П. Розподілені бази даних: навчальний посібник. К. ДУТ. 2018. 97с.

2. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань. Книга 1. Організація баз даних та знань. Львів : «Магнолія-2006». 2021. 440 с.

3. Codd, E. F. The Relational Model for Database Management, Addison-Wesley. 1990. pp. 371–388.

4. Гайджарж В., Ізварін І. Бази даних в інформаційних системах. Університет «Україна».2018. 418 с.

5. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань: навч. посібник. Львів : «Магнолія-2006». 2021. 584 с.

6. Blockchain basics: Introduction to distributed ledgers. URL: https://developer.ibm.com/learningpaths/get-started-blockchain/blockchain-basics/ (дата звернення 10.09.2023 р).

7. Amirya M. Blockchain technology in msme bookkeeping in Indonesia. Jurnal Ilmiah Akuntansi Peradaban. Vol. VIII No.2. 2022. P. 181-193.

8. Луцків А.М., Гладій В.В. Особливості функціонування та класифікації розподілених систем зберігання даних. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 455.

9. Луцків А.М., Гладій В.В. Структура та взаємодія між блоками у блокчейн. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 145.

10. Мартін Р. Чистий код. Створення і рефакторинг за допомогою Agile. В-во «Фабула». 2019. 448 с.

11. DB-Engines Ranking. DB-Engines. URL: https://db-engines.com/en/ranking/relational+dbms (дата звернення: 05.09.2023).

12. Allocation Fragmentation and Replication In Distributed Databases: A Quick Start Guide - Learn | Hevo. Learn | Hevo. URL: https://hevodata.com/learn/fragmentation-and-replication-in-distributed-database/ (дата звернення: 07.09.2023).

13. Distributed DBMS - Quick Guide. Online Tutorials Library. URL: https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_quick_guide.htm (дата звернення: 10.09.2023).

14. Raouf A. E. A., Badr N. L., Tolba M. F. Dynamic Distributed Database over Cloud Environment. Communications in Computer and Information Science. Cham, 2014. С. 67–76. URL: https://www.researchgate.net/publication/289052459_Dynamic_Distributed_Database_over_Cloud_Environment (дата звернення: 12.09.2023).

15. Install TensorFlow. URL: https://www.tensorflow.org/install (дата звернення 17.08.2023 р.).

16. TensorFlow Datasets: a collection of ready-to-use datasets. URL: https://www.tensorflow.org/datasets (дата звернення 17.08.2023 р.)

17. Cuda. URL: https://opencv.org/platforms/cuda/ (дата звернення 21.08.2023 р.)

18. NumPy v1.20 Manual. URL: https://numpy.org/doc/stable/ (дата звернення 25.08.2023р.).

19. Жидецький В.Ц. Охорона праці користувачів комп´ютерів. Львів: Афіша, 2011. 176 с.

20. Желібо Е.Н. Безпека життєдіяльності: Навчальний посібник/ За редакцією Е.П. Желібо, В.М. Пічі. – Київ: «Караве-ла», Львів: «Новий світ - 2000», 2011. 320с.

21. Стадник І.Я., Зварич Н.М. Оцінка хімічної обстановки при аваріях на хімічно небезпечних об’єктах викидом (виливом) небезпечних хімічних речовин та застосуванні хімічної зброї. ТНТУ. 2020. 36 С.

Додаток А

Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Міжнародний університет цивільної авіації (Марокко)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

XII Міжнародної науково-практичної конференції молодих учених та студентів
6-7 грудня 2023 року

УКРАЇНА
ТЕРНОПІЛЬ – 2023

Матеріали XII Міжнародної науково-практичної конференції молодих учених та студентів «АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль, 6-7 грудня 2023 року

УДК 004.031

А. М. Луцків канд. техн. наук, доцент, В. В. Гладій
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ ТА КЛАСИФІКАЦІЇ РОЗПОДІЛЕНИХ СИСТЕМ ЗБЕРІГАННЯ ДАНИХ

A. M. Lutskiv PhD., Assoc. Prof., V. V. Hladii

FUNCTIONING FEATURES AND CLASSIFICATION OF DISTRIBUTED DATA STORAGE SYSTEMS

Загалом, розподілені бази даних можна класифікувати за середовищем реалізації та функціонуванням на гомогенні (однорідні) та гетерогенні (неоднорідні). Структуру такої класифікації показано на рис 1.

Рисунок 1. Види середовищ розподілених баз даних

У гомогенних РБД усі вузли, між якими розподілені дані, використовують одні і ті ж СКБД та ОС. Основними властивостями однорідних РСКБД є: вузли розподіленої системи використовують подібне або однакове програмне забезпечення, зокрема це стосується СКБД; кожен вузол системи володіє інформацією про інші вузли та комунікує з ними при опрацювання користувацьких запитів; забезпечення доступу до даних виконується через спільний інтерфейс, що емулює роботу з єдиною БД. Існує два різновиди гомогенних РБД: автономна та неавтономна. У випадку автономної РБД, кожна БД, що входить до її складу функціонує незалежно від інших. Інтеграцію таких БД забезпечує зовнішня програмна система управління передачею повідомлень при обміні даними і їх станами. Неавтономна РБД передбачає розподіл даних між гомогенними вузлами, а центральна СКБД координує оновлення даних у вузлах розподілу. У неоднорідній РБД на різних вузлах встановлені різні операційні системи, використовуються різні СКБД з різними моделями даних.

До основних властивостей гетерогенних РБД належать:

– різні вузли містять різні схеми БД та прикладне ПЗ, тобто до складу системи можуть входити різні СКБД;

– опрацювання запитів ускладнене у зв'язку з різними схемами організації даних;

– опрацювання транзакцій вимагає значних ресурсів як апаратних, так і програмних;

– вузли можуть не знати про фрагменти розподіленої системи, тому комунікація при опрацюванні користувацьких запитів є обмеженою.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

УДК 004.031
Андрій Луцків канд. техн. наук, доцент, Віктор Гладій
Тернопільський національний технічний університет імені Івана Пулюя

СТРУКТУРА ТА ВЗАЄМОДІЯ МІЖ БЛОКАМИ У БЛОКЧЕЙН

Andriy Lutskiv PhD., Assoc. Prof., Viktor Hladii

STRUCTURE AND INTERACTION BETWEEN BLOCKS IN BLOCKCHAIN

Будь-яку структуру даних, яка використовується для зберігання інформації, можна вважати базою даних. Технологія блокчейн за своєю суттю — це не більше, ніж журнал для зберігання інформації про транзакції. До цього моменту блокчейни можна вважати базами даних.

Дані зберігаються у вигляді підписаних блоків, які зв'язуються один з одним, створюючи ланцюжок незмінних взаємопов'язаних записів даних. В загальному випадку, блоки у блокчейні можна представити як показано на рис. 1.

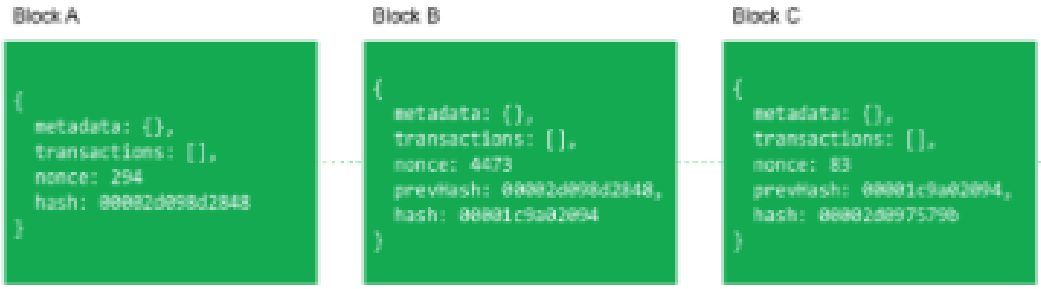

Рисунок 1. Блоки у блокчейні

Щоб підписати новий блок, вузол повинен знайти підпис SHA-256, який відповідає певним критеріям. Для цього він використовуватиме поле nonce для перебору можливих рішень.

Будь-який новий блок потрібно перевірити більшістю вузлів, що утворюють блокчейн. Після перевірки блоку він додається до всіх вузлів блокчейну. Цей спосіб перевірки нових блоків називається підтвердженням роботи (PoW) і був дуже поширеним на початку розвитку технології блокчейн.

Нині з'явилися інші методи підтвердження, такі як підтвердження частки (PoS). Якщо будь-яка інформація в даних усередині блоку змінена, підпис стає недійсним. Щоб знову зробити блок дійсним, цей підпис потрібно змінити. Щоб переконатися, що наступні блоки все ще працюють, для кожного з них також потрібно створити новий підпис. Навіть якщо вузол зможе відновити ці підписи, зміни мають бути прийняті більшістю вузлів, на яких розміщено блокчейн. З цих причин блокчейни є незмінними.

Жодна інформація, що міститься в даних блоків, не може бути змінена. Ними також керує набір децентралізованих вузлів, що усуває потребу в центральному органі для контролю всіх транзакцій. Через цю незмінність блокчейни набули популярності в таких галузях, як фінанси та нерухомість.