РОЗДІЛ ІІ. ІНФОРМАЦІЙНО-КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

DOI: 10.25140/2411-5363-2022-3(29)-94-101
УДК 004.658:004.4'41:004.82

Ірина Білоус[1], Дмитро Петренко[2], Олег Єрмоленко[3]

[1]кандидат технічних наук, доцент, завідувач кафедри інформаційних технологій та програмної інженерії Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: iryna.bilous.it@gmail.com. **ORCID:** https://orcid.org/0000-0002-5887-1955. **ResearcherID:** V-7697-2017

[2]аспірант кафедри інформаційних технологій та програмної інженерії
Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: dino96gr@gmail.com

[3]магістрант кафедри інформаційних технологій та програмної інженерії
Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: yermolenkooleg2@gmail.com

РОЗРОБКА ТА СУПРОВОДЖЕННЯ СИСТЕМ ІЗ РОЗПОДІЛЕНИМИ БАЗАМИ ДАНИХ НА ОСНОВІ ТЕХНОЛОГІЙ POLYGLOT PERSISTENCE

Проведено аналіз сучасних методів та технологій, що застосовуються для вирішення задач створення та супроводу розподілених баз даних із багатоваріантним збереженням. Обґрунтовано використання теореми Брюера стосовно обмежень та компромісних рішень щодо властивостей: узгодженості даних, доступності, допустимості поділу. Враховується розширення теореми Брюера щодо компромісу між затримками та узгодженістю при реплікації та принципи щодо базової доступності, гнучкого стану та кінцевої узгодженості. Наведено приклади архітектурних рішень для організації доступу в системі електронної комерції з бізнес-логіки та з єдиним механізмом доступу для мультимодельних СУБД. Показано недоліки підходів і можливості подальших досліджень.

Ключові слова: *розподілені бази даних; горизонтальне масштабування; polyglot persistence; мультимодельні СУБД; теорема Брюера.*

Рис.: 3. Бібл.: 12.

Актуальність теми дослідження. На сьогодні використання баз даних із багатоваріантною персистентністю (polyglot persistence), що передбачає доступ до кількох серверів БД з різними логічними моделями, набуває особливої популярності. Для збереження даних і вирішення різних задач з їх оброблення навіть у межах однієї інформаційної системи доводиться використовувати декілька різних СУБД, кожна з яких підтримує свою модель даних. Сучасні програмні рушії баз даних створені для виконання операцій із певними структурами даних і обсягами даних, наприклад, для роботи з наборами даних або сховищем, дуже швидкого отримання ключів і їхніх значень, або зберігання розширених документів чи складних графів інформації.

Постановка проблеми. У випадку забезпечення єдиного доступу до баз даних з багатоваріантною персистентністю з бізнес-логіки обсяг коду, що виконує збереження даних, зростає пропорційно числу СУБД, що використовуються; обсяг коду, що синхронізує дані, – пропорційний квадрату цього числа. Кратно числу використовуваних СУБД зростають витрати на забезпечення enterprise-характеристик (масштабованості, відмовостійкості, високої доступності) кожної з використовуваних СУБД. Неможливо забезпечити enterprise-характеристики підсистеми зберігання загалом – особливо транзакційність.

Для подолання проблем доступу до polyglot persistence БД із бізнес-логіки (відсутність гарантії відмовостійкості, громіздкість коду, необхідність інтеграції даних, дублювання даних, різний час для синхронізації даних) запропоновано рішення у вигляді мультимодельних СУБД з єдиним механізмом доступу.

Аналіз досліджень і публікацій. Проблемі ефективного зберігання гетерогенних даних БД, побудованих на різних моделях, присвячено багато робіт. Використання polyglot persistence вирішує проблему з обробкою гетерогенних даних, а системи, побудовані на її основі, відрізняються високою продуктивністю та низьким часом реакції на запит для кожної окремої моделі даних [1].

Водночас існує багато публікацій щодо можливих конфігурацій розподілених баз даних щодо досягнення базових показників [2]. Існують суперечності стосовно того, які з показників обрати провідними, а якими умовно можна знехтувати.

Найбільший важливим рішенням у цьому випадку є те, чи віддати перевагу доступності чи узгодженості, як описано в теоремі Брюера. У [3] наведено порівняльну модель, яка співвідносить показники, що сприяє вибору сховища даних, відповідно до вимог певної інформаційної системи. У роботі [4] вивчено, як узгодженість реалізується в різних NoSQL базах даних та описано стратегії для забезпечення узгодженості та змінні компроміси між узгодженістю та іншими атрибутами якості, такими як доступність, затримка та допустимість поділу. Балансування може застосовуватися для різних потреб залежно від системних вимог і бажаного результату, при цьому існують балансувальники для різних рішень баз даних [5; 6]. Велика проблема існуючих рішень полягає в тому, що типи БД мають бути однаковими.

Виділення недосліджених частин загальної проблеми. Не вирішеним завданням є розробка й супроводження розподілених мультимодельних СУБД з єдиним механізмом доступу, що забезпечують основні показники розподілених систем на основі теореми Брюера. У цій роботі особливий акцент робиться на те, що розподілена база даних представляється в вигляді єдиної системи, до складу якої входять вузли (сервери БД), що реалізують різні моделі даних.

Необхідно виконати аналіз властивостей теореми Брюера (та її розширення) стосовно до розподілених баз з єдиним механізмом доступу до багатоваріантних даних, вибору архітектурних рішень при побудові та супроводженні високонавантаженої інформаційної системи електронної комерції для проведення подальших експериментальних досліджень з метою отримання компромісних рішень за властивостями базової доступності й доступності поділу. За основу отримання таких рішень взяти властивість кінцевої узгодженості як незмінюваний параметр.

Мета статті. Метою цієї роботи є аналіз і проєктування розподілених баз даних на основі технологій багатоваріантної персистентності, а також формалізація властивостей теореми Брюера та її розширення стосовно до розподілених мультимодельних баз даних.

Виклад основного матеріалу. При використанні сучасних баз даних у високонавантажених інформаційних системах насамперед розглядається використання механізмів масштабованості в двох напрямках.

Простішим, але найбільш затратним варіантом є нарощування апаратних потужностей сервера БД, це так зване вертикальне масштабування. Другий напрям використовує горизонтальне масштабування, при якому виконується додавання кількості серверів та розподілення бази даних між ними.

Особливістю вертикальної масштабованості є присутність деякого порогу (або межі) і висока вартість витрат на реалізацію поставлених вимог в області підвищення продуктивності сервера. Тому найбільш прийнятним варіантом вирішення більшості завдань у розподілених системах є горизонтальне масштабування. Цей механізм масштабування передбачає збільшення кількості серверів при зростаючих навантаженнях під час обробки великих обсягів запитів [10].

Ці варіанти можуть допомогти вирішити проблеми високого навантаження, але без оптимізації структури бази даних рентабельність процесу буде досить низькою. Така оптимізація з цілями масштабування може відбуватися трьома основними шляхами: партиціювання, шардінг, реплікація [11].

Однак горизонтальне масштабування системи управління даними є досить складним завданням, оскільки традиційні SQL-орієнтовані сервера погано адаптовані до повної фрагментації даних. Це призводить до розробки нових концепцій у роботі розподілених баз даних і створення підходів, які вирішують проблеми горизонтального масштабування.

З іншого боку, як шляхи вирішення проблем реляційних БД запропоновані нові технології персистентності – open source рішення NoSQL – нереляційні розподілені бази даних із підтримкою гнучкої схеми бази даних і горизонтальної масштабованості, у яких як мова запитів не використовується SQL [1].

У реальних інформаційних системах використовується поєднання реляційних та NoSQL рішень, серед яких документні (document), ключ-значення (key-value), графові (graph) і колоночні сховища (column).

Рекомендації щодо типу бази даних для використання на основі функціональності даних на прикладі системи електронної комерції показано на рис. 1.

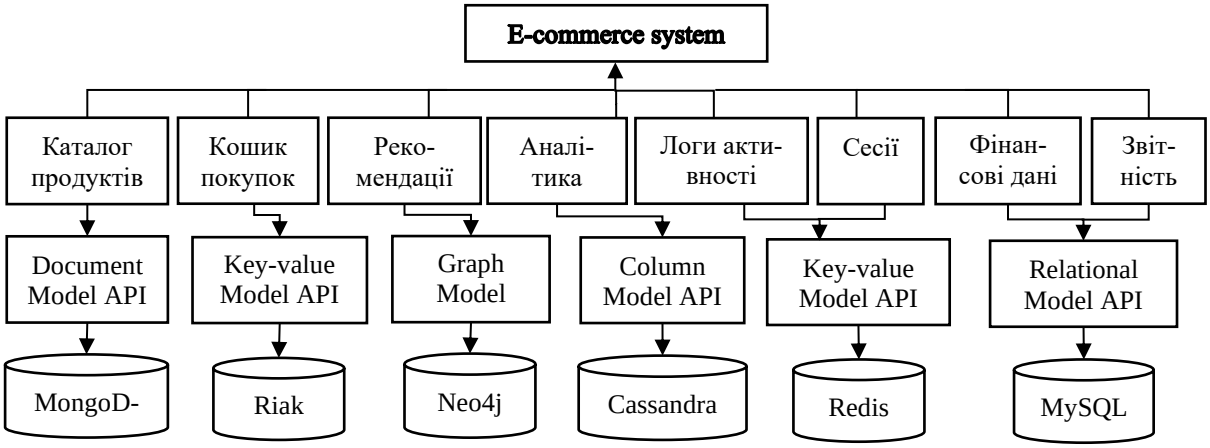


Рис. 1. Схема реалізації системи електронної комерції на боці сервера на основі polyglot persistence

Для даних каталогу продукції, з якими проходить багато читань і мало записів, і самі записи про продукти утворюють природні агрегати найкраще обрати документну модель даних. Дані з кошику покупок потребують високої доступності в багатьох місцях, дані можуть об’єднувати неузгоджені записи. Найкраще підходить документна модель даних, можливо, модель ключ-значення.

Для даних про рекомендації, де необхідний швидкий перехід за посиланнями між покупками продуктів друзів, і рейтингами, варто обрати графову модель або колоночне сховище.

У випадку використання широкомасштабної аналітики потрібна організація даних у великому кластері, тому найкраще обрати колоночне сховище.

Журнали активності користувачів та аналіз соціальних мереж потребують великої кількості записів на кількох вузлах. Для цього найкраще підійде модель ключ-значення або документна модель. Вимагається швидкий доступ для читання та запису при використанні даних про сеанси користувача, немає необхідності зберігати довговічну інформацію. У такому випадку найкраще підійде модель ключ-значення, можлива реалізація і з документною моделлю.

Використання фінансових даних потребує оновлення транзакцій, в цьому випадку таблична структура відповідає даним, тому найкраще обрати реляційну модель. Так само для даних щодо звітності – SQL добре взаємодіє з інструментами звітності, але можлива реалізація і колоночної моделі.

Мультимодельні СУБД

Модель багатоваріантної персистентності використана і при побудові мультимодельних СУБД, які спрямовані вирішити проблеми з доступом до різних моделей із бізнес-логіки.

Мультимодельні СУБД спрямовані на об'єднання різних логічних моделей даних в єдиний інтегрований програмний рушій, який використовує уніфіковану мову запитів і надає єдиний API, який використовується в усіх логічних моделях даних. Основна концепція полягає в тому, щоб на фізичному рівні зберігати всі дані в одній СУБД, а потім для обробки представляти інші моделі шляхом зіставлення моделей більш високого рівня з поданням нижчого рівня [1].

Приклад такої реалізації (з найбільшою кількістю підтримуваних API моделей) для тієї ж задачі побудови системи електронної комерції представлено на рис. 2.

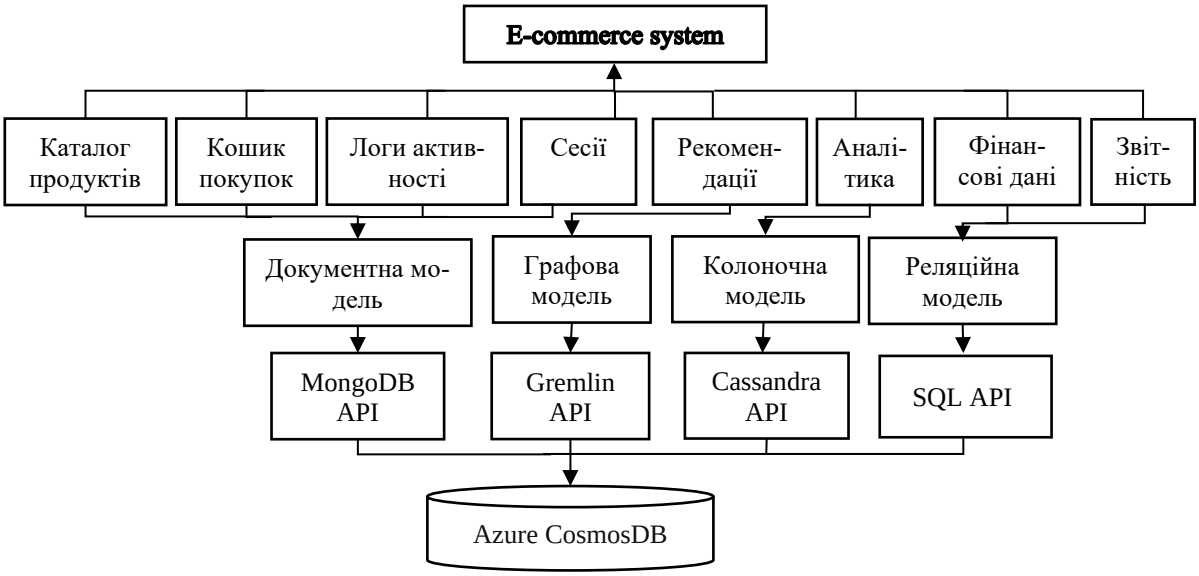


Рис. 2. Схема реалізації системи електронної комерції на стороні сервера на основі мультимодельної СУБД

Відомі лідери серед мультимодельних СУБД:

- на основі реляційної моделі, що реалізують графову та документну модель (Oracle, MS SQL, PostgreSQL);
- на основі документної моделі, що реалізують графову й реляційну (MarkLogic), графову і модель «ключ-значення» (MongoDB);
- на основі моделі «ключ-значення», що реалізують документну і графову модель (Redis);
- на основі колоночної моделі, що реалізують документну і графову модель (DataStax);
- без основної моделі, що реалізують графову і документну модель (ArangoDB, OrientDB), та графову, колоночну, документну і реляційну модель (Azure CosmosDB).

Найбільш широка мультимодельність в Azure CosmosDB на сьогодні являє собою лише можливість використання декількох баз даних, що підтримують різні моделі від одного виробника (Azure), що не вирішує всіх проблем багатоваріантного збереження. Обрана користувачем модель даних та API, що використовується, фіксується в момент створення акаунта в сервісі. Неможливо отримати доступ до даних, що завантажені в одній моделі, в форматі іншої моделі.

Теорема Брюера та її розширення

Серед властивостей, яким повинна задовольняти розподілена мультимодельна база даних, виділимо наступні [12].

–Локальна автономність. Локальні дані належать локальним вузлів і управляються адміністраторами локальних БД.

–Безперервне функціонування. Видалення або додавання вузла не повинно вимагати зупинки системи в цілому.

–Незалежність від фрагментації. Доступ до даних не повинен залежати від наявності або відсутності фрагментації і від типу фрагментації.

–Обробка розподілених транзакцій. Протокол обробки розподіленої транзакції має забезпечувати виконання чотирьох основних властивостей транзакції: атомарність, узгодженість, ізольованість та довговічність (ACID).

–Обробка розподілених запитів. Система повинна автоматично визначати методи виконання з'єднання (об'єднання) даних.

–Незалежність від типу СУБД. Система повинна бути здатною функціонувати поверх різних локальних СУБД, можливо, з різними моделями даних (вимога гетерогенності).

Неможливість підтримки наведених критеріїв розподіленості (особливо ACID) якоюсь мірою компенсована евристичним принципом, відомим як теорема Брюера або теорема CAP [7]. Дана теорема використовується для обґрунтування компромісів при проєктуванні розподілених систем і ґрунтується на тому, що побудова інформаційних систем або розподілених структур не може виконуватися при одночасному виконанні наступних властивостей: узгодженості (consistency), доступності (availability) і стійкості до розділення (partition tolerance) [10].

Згідно CAP, всі розподілені СУБД забезпечують тільки дві з трьох властивостей – узгодженість (consistency) даних, доступність (availability) і доступність поділу (partition tolerance). Відповідно:

− consistency – наявності тільки однієї копії даних, яка відповідає останній за часом операції оновлення;

− availability – доступність даних при наявності операцій оновлення;

− partition tolerance – здатність вузлів, між якими немає зв'язку, продовжувати працювати незалежно один від одного.

Грані CAP (рис. 3а) можна описати так.

PA – у випадку мережного поділу вузлів системи (горизонтального масштабування), вона продовжує відповідати користувачам на запити, не гарантуючи узгодженості даних.

PC – у випадку мережного поділу вузлів системи, вона припиняє відповідати користувачам на запити, дані залишаються узгодженими.

CA – у разі відсутності мережного поділу дані доступні та узгоджені (нормальний режим роботи).

До прикладу, такі системи, як Cassandra та Riak, можуть забезпечити доступність, тоді як такі системи, як HBase, MongoDB і DynamoDB забезпечують надійну узгодженість.

Виходячи з означення, доступність в CAP, має дві суттєві проблеми. Перша – нема поняття часткової доступності, чи якоїсь її міри, а є тільки повна доступність. Інша проблема – необмежений час відповіді на запити, навіть якщо система відповідає годину, вона все ще доступна.

Відповідно до означення доступності "...every node (if not failed) always..." розділення в мережі не включає в себе вузли, що перестали працювати. Якщо сервер на якийсь час вийшов з ладу, він може відновитись, обмінятися даними з іншими вузлами та продовжити роботу. В випадку розділення в мережі – необхідно чекати відновлення з'єднання. Тому варто звертати увагу на здатність системи відновлюватись, але за рамками CAP теореми.

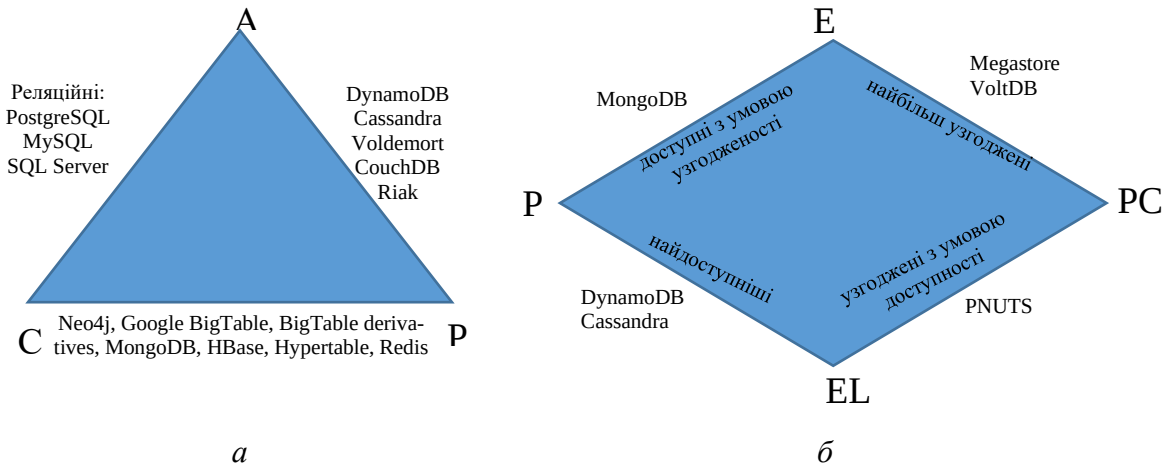
Теорема PACELC [8] – розширення теореми CAP, яке свідчить, що у разі розділення в мережі (P) у розподіленій базі даних необхідно вибирати між доступністю (A) та узгодженістю (C) (згідно з теоремою CAP), але в будь-якому випадку, навіть якщо система працює нормально без поділу (E), потрібно вибирати між затримками (L) і узгодженістю (C).

В випадку горизонтального масштабування в розподілених базах даних вимога високої доступності передбачає, що система має реплікувати дані. Поки розподілена система реплікує дані, виникає потреба вибирати між узгодженістю та затримками.

Вся теорема зводиться до твердження: IF P -> (C or A), ELSE (C or L).

Затримка (latency) – це час, за який клієнт отримає відповідь і який регулюється будь-яким рівнем consistency. Latency, у певному сенсі є ступенем доступності.

Грані PACELC та приналежність певних СУБД показана на рис. 3, *б*.



а *б*

Рис. 3. Ілюстрація взаємозв'язку:
а – CAP і популярних СУБД; б – PACELC та популярних СУБД

Чисті PC системи можуть взагалі бути не доступні, оскільки намагатимуся прийти до узгодженого стану й не відповідатимуть. PC системи можуть дати не очікувану строгу узгодженість (strong consistency), а кінцеву (eventual consistency) з певним порогом. У цьому контексті краще використовувати принципи BASE [9].

BASE – це своєрідний контраст ACID, який говорить, що справжня узгодженість не може бути досягнута в реальному світі й не може бути змодельована в системах, що високомасштабуються. Запропоновані уточнення властивостей.

Базова доступність (basic availability). Система відповідає на будь-який запит, але ця відповідь може містити помилку або неузгоджені дані.

Гнучкий стан (soft-state). Стан системи може змінюватися згодом через зміни кінцевої узгодженості.

Кінцева узгодженість (eventual consistency). Система, зрештою, стане узгодженою. Система продовжуватиме приймати дані та не перевірятиме кожну транзакцію на узгодженість.

Перспектива подальших досліджень. Ця робота є початковим етапом у дослідженні одного з напрямів використання розподілених баз даних із багатоваріантною персистентністю. Подальша робота буде полягати у створенні мультимодельної СУБД, що підтримує динамічне використання різних моделей даних та в проведенні навантажувальних впливів на такі бази даних із використанням теореми Брюера та її розширення. При цьому передбачається отримати кількісні та якісні оцінки компромісних рішень щодо властивостей базової доступності і допустимості поділу для фіксованого параметра властивості кінцевої узгодженості.

Висновки. У статті:

- виконано огляд деяких сучасних рішень, що забезпечують розробку та супроводження розподілених СУБД на основі багатоваріантного збереження. Ці рішення ґрунтуються на використанні теореми Брюера (CAP) і її розширення (PACELC) та принципах BASE;

- відповідно до властивостей теорем наведено формалізація визначень для розподілених баз даних із багатоваріантною персистентністю;

- описано проблему побудови високонавантаженої інформаційної системи електронної комерції з розподіленою динамічною базою з єдиним механізмом доступу до багатоваріантних даних, вибору її конфігурації для проведення подальших експериментальних досліджень з метою отримання компромісних рішень за властивостями.

Список використаних джерел

1. Information Technology of Supporting Architectural Solutions Using Polyglot Persistence Concept in Learning Management Systems / O. Arsirii, M. Glava, Kolonko M., A. Glumenko // Applied Aspects of Information Technology. – 2020. – Vol. 3, № 2. – Pp. 13-31.

2. Мухін В. Є. Аналіз ефективності оброблення запитів серверами гетерогенних розподілених баз даних / В. Є. Мухін, Я. І. Корнага // Технічні науки та технології. – 2016. – № 1(3). – С. 89-94.

3. NoSQL database systems: a survey and decision guidance / F. Gessert, W. Wingerath, S. Friedrich et al. // Comput Sci Res Dev. – 2017. – Vol. 32. – Pp. 353-365.

4. Diogo M. Consistency Models of NoSQL Databases / M. Diogo, B. Cabral, J. Bernardino // Future Internet. – 2019. – Pp. 43-51.

5. Joshi S. Balanced Load in Distributed System with NoSQL Middleware / S. Joshi, S. Ameta, G. Lavania // International Journal of Emerging Technologies and Innovative Research. – 2019. – Vol. 6(5). – Pp. 133-137.

6. Rukkas K. Load balancing consistency in a distributed datastore / K. Rukkas, G. Zholtkevych // Системи управління, навігації та зв'язку. – 2020. – Т. 2 (60). – С. 95-100.

7. Brewer E. A. Towards robust distributed systems [Electronic resource] / E. A. Brewer // The nineteenth annual ACM symposium. – ACM Press, 2000. – Accessed mode: https://doi.org/10.1145/343477.343502.

8. Wojciech Golab. Proving PACELC / Wojciech Golab // ACM SIGACT News. – 2018. – Vol. 49, Issue 1. – Pp. 73-81. – doi:10.1145/3197406.3197420.

9. Pritchett D. Base an acid alternative / D. Pritchett // ACM Queue. – 2008. – Vol. 6. – Pp. 48-55.

10. Гречанінов В. Ф. Аналіз і проектування розподілених систем на основі кластерних технологій / В. Ф. Гречанінов // Кібербезпека: освіта, наука, техніка. – 2022. – № 2(14). – Pp. 186-191. – DOI: https://doi.org/10.28925/2663-4023.2021.14.186191.

11. Нагорний П. В. Проблеми масштабування баз даних у високонавантажених системах [Електронний ресурс] / П. В. Нагорний // Новітні технології сучасного суспільства (НТСС-2020) : збірник тез доп. І Міжнар. наук.-практ. конф. (м. Чернігів, 17 грудня 2020 р.). – Чернігів : Нац. ун-т «Чернігівська політехніка», 2020. – С. 175-177. – Режим доступу: https://eportfolio.kubg.edu.ua/data/conference/6390/document.pdf.

12. Бальченко І. В. Проблеми розроблення неоднорідних розподілених систем управління базами даних / І. В. Бальченко // Технічні науки та технології. – 2017. – № 2(4). – С. 67-71.

References

1. Arsirii, O., Glava, M., Matthias Kolonko, & Glumenko, A. (2020). Information Technology of Supporting Architectural Solutions Using Polyglot Persistence Concept in Learning Management Systems. *Applied Aspects of Information Technology*, *3*(2), 13–31.

2. Mukhin, V. Ye., & Kornaha, YA. I. (2016). Analiz efektyvnosti obroblennia zapytiv serveramy heterohennykh rozpodilenykh baz danykh [Efficiency analysis of requests' processing by the server inthe heterogeneous distributed databases]. *Tekhnichni nauky ta tekhnolohii – Technical sciences and technologies*, *1*(3), 89-94.

3. Gessert, F., Wingerath, W., Friedrich, S. et al. (2017). NoSQL database systems: a survey and decision guidance. *Comput Sci Res Dev*, *32*, 353-365.

4. Diogo, M., Cabral, B., Bernardino, J. (2019). Consistency Models of NoSQL Databases. Future Internet (pp. 43-51).

5. Joshi, S., Ameta, S., & Lavania, G. (2019). Balanced Load in Distributed System with NoSQL Middleware. *International Journal of Emerging Technologies and Innovative Research*, *6*(5), 133-137.

6. Rukkas, K., & Zholtkevych, G. (2020). Load balancing consistency in a distributed datastore. *Systemy upravlinnia, navihatsii ta zviazku – Control, navigation and communication systems, 2*(60), 95-100.

7. Brewer, E. A. (2000). Towards robust distributed systems. The nineteenth annual ACM symposium. ACM Press. https://doi.org/10.1145/343477.343502.

8. Wojciech Golab. (2018). Proving PACELC. *ACM SIGACT News*, *49*(1), 73–81. doi:10.1145/3197406.3197420.

9. Pritchett, Dan. (2008). Base an acid alternative. *ACM Queue, 6*. 48-55. 10.1145/1394127.1394128.

10. Grechaninov, V. (2022). Analiz i proektuvannia rozpodilenykh system na osnovi klasternykh tekhnolohii [Analysis and design of distributed systems based on cluster technologies]. *Kiberbezpeka: osvita, nauka, tekhnika – Cybersecurity: Education, Science, Technique*, (2(14)), 186-191. https://doi.org/10.28925/2663-4023.2021.14.186191.

11. Nahornyi, P. (2020). Problemy masshtabuvannya baz danykh u vysokonavantazhenykh systemakh [Problems of database scaling in highly loaded systems]. *Novitni tekhnolohiyi suchasnoho suspilstva (NTSS-2020): zbirnyk tez dop. I Mizhnar. nauk.-prakt. konf. (Chernihiv, 17 december 2020) – Latest technologies of modern society (NTSS-2020): collection of abstracts. I International science and practice conf.* (pp. 175-177). Chernihiv Polytechnic National University. https://eportfolio.kubg.edu.ua/data/conference/6390/document.pdf.

12. Balchenko, I. (2017). Problemy rozroblennia neodnoridnykh rozpodilenykh system upravlinnia bazamy danykh [Issues of the development of heterogeneous distributed database management systems]. *Tekhnichni nauky ta tekhnolohii – Technical sciences and technologies*, (2(4)), 67-71.

Отримано 25.09.2022

UDC 004.658:004.4'41:004.82

Iryna Bilous[1], Dmytro Petrenko[2], Oleh Yermolenko[3]

[1]PhD in technical sciences, associate professor, head of Information technology and Software engineering department
Chernihiv National University of Technology (Chernihiv, Ukraine)
E-mail: iryna.bilous.it@gmail.com. **ORCID:** https://orcid.org/0000-0002-5887-1955. **ResearcherID:** V-7697-2017

[2]PhD student of Information technology and Software engineering department
Chernihiv National University of Technology (Chernihiv, Ukraine)
E-mail: dino96gr@gmail.com

[3]Master's Degree Student of Information Technology and Software Engineering Department
Chernihiv National University of Technology (Chernihiv, Ukraine)
E-mail: yermolenkooleg2@gmail.com

THE DEVELOPMENT AND MAINTENANCE OF DISTRIBUTED DATABASE SYSTEM BASED ON POLYGLOT PERSISTENCE TECHNOLOGIES

The analysis of modern methods and technologies used for solving the problems of creating and maintaining distributed databases with a multi-variant storage was carried out. Distributed databases with a single mechanism for accessing multi-variate data are singled out. The use of scalability mechanisms in the direction of horizontal scaling (in which a certain number of servers are increased within one system) is considered.

The justification for the use of the Brewer's theorem is given, in relation to restrictions and compromise solutions regarding properties: consistency, availability, partition tolerance. In relation to this theory, a formalization of the definition of properties for distributed multimodel databases is presented. The extension to the Brewer's theorem, which requires a trade-off between delays and consistency when applying a replication, and the principles for basic availability and eventual consistency are taken into account.

The ways to create and support a highly loaded e-commerce system with support for distributed databases with multi-variate persistence are provided in the article. Examples of possible architectural solutions for organizing access in such a system using a mix of relational and NoSQL (such as document, key-value, graph and column storage) are given. Such architectural solution as distributed multi-model DBMS with a single access mechanism is proposed to overcome the problems of accessing databases with multi-variant saving from business logic. The shortcomings for such solution's modern implementations and ways of its improvement are shown. In particular, it is a support for all known data models and dynamic access to data loaded in one model in the format of another model.

This work provides an opportunity to obtain and analyze experimental data at the next stages for the study of distributed multi-model dynamic databases using loading effects to obtain quantitative and qualitative characteristics of basic availability and partition tolerance for a fixed parameter of the eventual consistency for multimodel DBMS with a single access mechanism.

Key words: *distributed databases, horizontal scaling, polyglot persistence, multimodel DBMS, Brewer's theorem.*

Fig.: 3. References: 12.

Білоус І., Петренко Д., Єрмоленко О. Розробка та супроводження систем із розподіленими базами даних на основі технологій polyglot persistence. *Технічні науки та технології.* 2022. № 3(29). С. 94-101.