

А. В. Миргородський; О. В. Романюк, канд. техн. наук, доц.

РОЗВИТОК МЕТОДІВ СЕГМЕНТАЦІЇ ДАНИХ В РОЗПОДІЛЕНИХ СИСТЕМАХ КЕРУВАННЯ БАЗАМИ ДАНИХ

В статті розглянуто актуальну задачу підвищення ефективності роботи розподілених інформаційних систем шляхом використання методів сегментації (фрагментації) даних. Наголошено, що забезпечення балансу між доступністю, узгодженістю та швидкістю обробки запитів є одним із ключових завдань сучасних систем керування базами даних. Проаналізовано широкий спектр методів сегментації, включаючи діапазонну, хеш-, географічну, темпоральну сегментацію, мікро-шарди, сегментацію з урахуванням мережі та на основі знань. Для кожного підходу охарактеризовано принцип роботи, переваги та обмеження, а також специфіку застосування у різних сценаріях використання. На основі проведеної систематизації сформовано порівняльну таблицю, яка дозволяє оцінити методи за такими критеріями, як динамічність розподілу, адаптивність до змін, складність реалізації, апаратні вимоги та можливості локалізації даних. Запропоновано інтегральний критерій ефективності, що враховує динамічність розподілу, адаптивність, локалізацію та апаратні вимоги й дозволяє кількісно оцінити ефективність різних методів незалежно від їхньої концептуальної природи. Показано, що адаптивні методи забезпечують найкращі результати в умовах достатніх ресурсів, тоді як географічна сегментація та мережево орієнтовані підходи є більш ефективними у ресурсно обмежених середовищах. Проаналізовано вплив вагового коефіцієнта апаратних вимог на результат критерія, що дозволило сформувати рекомендації щодо вибору методу сегментації для різних інфраструктурних сценаріїв. Наукова новизна полягає у запропонованій інтегральній системі оцінювання ефективності методів сегментації даних, що поєднує показники динамічного розподілу, адаптивності, локалізації та апаратних вимог у єдиному нормалізованому критерії. Отримані результати становлять основу для подальших досліджень у напрямку розроблення універсальних адаптивних механізмів сегментації, інтеграції їх із сучасними моделями узгодженості даних та застосування алгоритмів машинного навчання для автоматичної класифікації сегментів.

Ключові слова: розподілені системи, бази даних, сегментація даних, шардинг, масштабованість, узгодженість даних, розподілене зберігання.

Вступ

У сучасних розподілених інформаційних системах одним із ключових завдань є забезпечення балансу між високою доступністю, узгодженістю та швидкістю обробки даних. Застосування методів сегментації або фрагментації даних дозволяє ефективно розподіляти навантаження між вузлами, що позитивно впливає як на продуктивність розподілених систем [1], так і може застосовуватись в різноманітних моделях узгодженості [2]. Більш старі підходи, такі як діапазонна чи хеш-сегментація, добре зарекомендували себе в традиційних сценаріях застосування, проте вони не враховують динамічні зміни в навантаженні та тренди в доступі до даних. Це може призвести до проблем із нерівномірним навантаженням, затримками під час обробки запитів та зниження ефективності реплікації [3].

Останні дослідження та напрацювання в цій сфері акцентують увагу на недоліках класичних методів сегментації та використовують часові, географічні чи поведінкові характеристики даних для підвищення продуктивності системи [4]. Проте, відсутність систематизованої класифікації та критичного порівняння цих методів ускладнює їх практичне впровадження та подальше покращення і розвиток засобів сегментації даних. Таким чином, актуальним завданням є узагальнення та аналіз сучасних методів сегментації даних у розподілених системах з метою виявлення їхніх переваг, недоліків і перспектив

подальшого розвитку.

Метою роботи є систематизація наявних підходів до сегментації даних, критичне порівняння їхніх характеристик та визначення напрямів удосконалення для підвищення ефективності роботи розподілених інформаційних систем.

Наукова новизна полягає у запропонованій інтегральній системі оцінювання ефективності методів сегментації даних, що поєднує показники динамічного розподілу, адаптивності, локалізації та апаратних вимог у єдиному нормалізованому критерії.

Методологічно робота спирається на систематизацію літератури, формалізацію порівняльних характеристик, побудову математичної моделі інтегрального показника, а також аналіз впливу коефіцієнта апаратних вимог на відносну ефективність методів сегментації у різних сценаріях застосування.

Аналіз та класифікація класичних методів сегментації даних

Забезпечення високої доступності та узгодженості даних постає важливим питанням в дизайні розподілених інформаційних систем, включаючи системи керування базами даних. Проте, в реальних сценаріях застосування також високий пріоритет має швидкодія обробки запитів. Тому, розподілені системи часто реалізують певний механізм сегментації даних на вузлах.

Сегментація даних (часто також фрагментація даних або шардинг даних, англ. Data Sharding) – це стратегія поділу даних на менші керовані сегменти, фрагменти або шарди, що можуть бути довільно розміщені на вузлах системи [5]. Розподілена СКБД без сегментації даних оперує загальним набором даних на рівні вузлів, наприклад, шляхом повної дуплікації всієї інформації на кожному з вузлів. Система з сегментацією керує значно меншими частинами й дозволяє гнучко налаштувати розподіл даних як на окремому вузлі, так і в усій системі в цілому. Наприклад, більш важливі сегменти даних можуть дублюватись на більшій кількості вузлів або ж мати певні пріоритети під час обробки даних всередині окремо взятого сервера.

Методи сегментації даних, як і інші комплексні механізми в розробці сучасного програмного забезпечення, можна класифікувати по-різному на основі цілого ряду критеріїв. Один з варіантів класифікації наведено на діаграмі на рис. 1.

Цей варіант класифікує методи за трьома напрямками: за етапом виконання сегментації даних, за стратегією розподілу даних в окремі фрагменти (або визначення місцезнаходження для нових даних), а також за типом розподілу структури даних під час сегментації. Розглянемо більш детально види сегментації згідно з цієї класифікації.

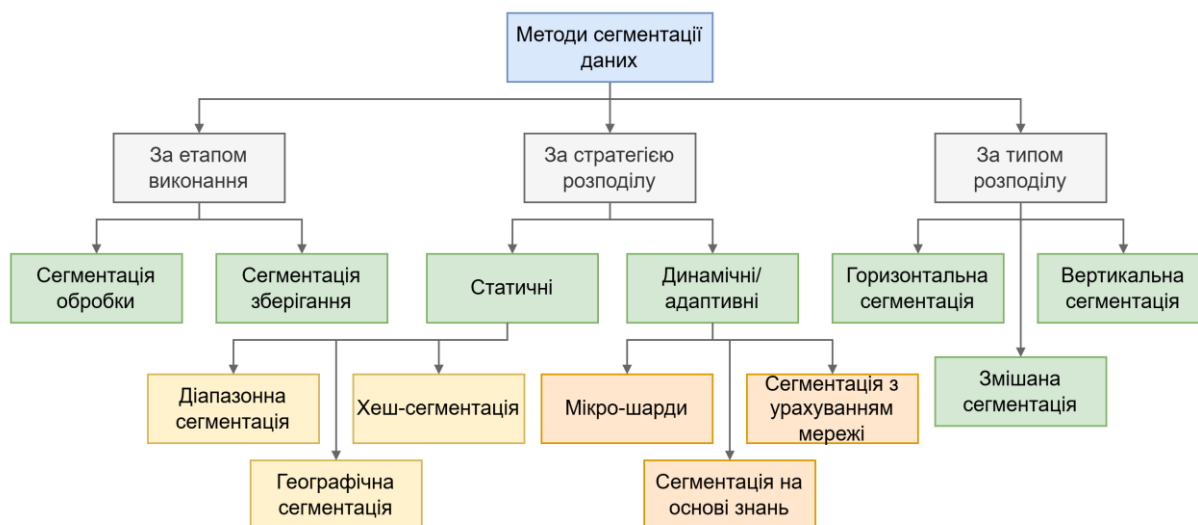


Рис. 1. Класифікація методів сегментації даних

За етапом виконання варто розрізняти дві основні складові сегментації даних – це сегментація обробки та сегментація зберігання. Сегментація обробки (англ. Processing Sharding) є більш простим методом, що не передбачає довільного розміщення фрагментів на вузлах, а лише розподіляє між декількома серверами виконання комплексного запиту чи транзакції [6]. Це дозволяє значно підвищити швидкість обробки запитів без ускладнення архітектури розподіленої системи. Проте, покращення швидкодії варто очікувати лише для систем з високим об'ємом запитів вичитування даних.

Сегментація зберігання (англ. Storage Sharding, іноді State Sharding) вже передбачає розподіл фрагментів між різними вузлами системи [7]. Наприклад, розподілена СКБД може зберігати різні рядки однієї таблиці на декількох фізичних серверах або ж відокремити окрему підмножину атрибутів для аналітичних запитів на окремому вузлі. Такий тип сегментації поширений також і в інших типах розподілених систем, наприклад, блокчейн. Застосування сегментації зберігання також вимагає і сегментації обробки, адже необхідні для виконання запиту дані можуть знаходитись в сегментах на різних серверах.

Можливість контролювати окремі сегменти даних (включаючи правила їх реплікації, як-от кількість копій та їх розміщення) дозволяє також підвищити швидкість обробки операцій запису. Оновлення лише декількох сегментів відбувається значно швидше, ніж синхронізація всіх вузлів в класичній розподіленій системі без сегментації. Тому, сегментація зберігання пропонує кращу швидкість в більшому спектрі сценаріїв застосування. По цій причині під сегментацією даних зазвичай розуміють і розподілене зберігання фрагментів, і розподілену обробку даних в них.

За стратегією розподілу методи сегментації зберігання зазвичай використовують простий розподіл на основі статичних правил. Зазвичай це розподіл на основі ключового атрибуту таблиці або документу (у випадку NoSQL систем). Більш поширеними є способи розподілу на основі діапазону значень та на основі хешування. Діапазонна сегментація передбачає пряму класифікацію записів на основі значення ключа [8]. Наприклад, якщо ключем таблиці є цифровий ідентифікатор, то перший фрагмент може містити записи з ключем від 0 до 999, наступний – від 1000 до 1999 і так далі. Цей спосіб дуже простий в реалізації та надає передбачуваний розподіл, що добре підходить для транзакцій, що використовують дані з одного діапазону. Проте, необхідно враховувати патерни доступу до даних: якщо певний діапазон використовується частіше за інші, то це створить нерівномірне навантаження на вузли розподіленої системи.

Сегментація за допомогою хешування застосовує той же принцип, що і в звичайних хеш-таблицях – ідентифікатор запису перетворюється за допомогою певної хеш-функції і отримане значення визначає до якого саме сегменту цей запис буде розподілено [9]. Це дозволяє досягти більш рівномірного розміщення даних на вузлах і добре підходить для систем з рівномірним доступом, але окремі вузли все-одно можуть бути більш навантаженими в випадку частого доступу до одного й того самого запису.

Ще одним популярним методом розподілу даних між шардами є географічна сегментація, що частіше застосовується в глобальних розподілених системах зі значною кількістю територіально відокремлених вузлів. Такий тип сегментації повинен також враховувати відносно розміщення серверів та фрагментів на них до кінцевих користувачів або клієнтів інформаційної системи, що дозволить зменшити затримки при обробці даних [10]. Крім того, це дозволяє застосовувати додаткові регіональні обмеження та вимоги до зберігання даних на серверах. Основними проблемами географічної сегментації є підвищені затримки під час виконання запитів, що вимагають дані з інших регіонів, або ж втрата чи погіршення переваг сегментації за невдалого розміщення вузлів системи [11].

Варто також розглянути класифікацію за типом розподілу. В класичних реляційних базах даних зазвичай використовується горизонтальна сегментація, коли кожен запис зберігає всі свої атрибути й кожен шард містить просто певний набір цих записів [12]. В такому випадку

непрацездатність одного з вузлів розподіленої системи може привести до повної втрати доступу до окремих наборів даних, але це також спрощує та пришвидшує виконання запитів, що потребують доступу до великої кількості атрибутів.

Вертикальна сегментація розподіляє атрибути однієї структури даних на декількох вузлах зі збереженням ключа або іншого методу ідентифікації записів [13]. Наприклад, один фрагмент може містити ПІБ користувачів, коли-як інший буде зберігати вік або адресу. Такий вид сегментації робить обробку типових запитів з великою кількістю атрибутів більш складною, але він добре підходить для обробки комплексних аналітичних запитів [14], тому іноді застосовується в спеціалізованих реляційних БД.

Також іноді можна зустріти змішану сегментацію, що за принципом формування фрагментів є більш подібною до вертикальної, але має іншу логіку виконання. Зазвичай під час застосування змішаної фрагментації спочатку визначаються критерії для горизонтального поділу – наприклад, на основі географічних даних в записах. Після цього отримані сегменти додатково підрозділяються за принципами вертикальної фрагментації на декілька менших сегментів з різними наборами атрибутів [12]. Такий підхід дозволяє створювати спеціалізовані вузли розподіленої системи, чиї фрагменти даних можуть самостійно опрацьовувати певні типи запитів, що дозволяє зменшити потребу в додатковій комунікації з іншими серверами.

Розвиток динамічних методів сегментації

Класичні методи сегментації не враховують додаткові характеристики даних під час розміщення фрагментів, що й призводить до потенційного нерівномірного навантаження системи. Єдине виключення – це географічна сегментація, але її сфера застосування обмежена й дає покращення лише на великих розподілених системах. Тому, дедалі більшого розвитку й поширення отримують динамічні методи сегментації даних.

Основна характеристика динамічних методів сегментації даних подібна до властивостей нових моделей узгодженості – розподілена система повинна адаптуватись до змін в даних та патернах доступу до них [15]. Одне з досліджень пропонує метод темпоральної сегментації, що повинен враховувати часові характеристики даних [16].

В цьому методі замість діапазонної функції або хеш-функції використовується окремий алгоритм кластерингу даних, що враховує дату створення та оновлення записів, частоту запитів та відслідковує потенційні тренди або ж циклічність в патернах доступу. Базуючись на цьому, алгоритм може класифікувати кожен запис системи (наприклад, розподіленої СКБД) в одну з трьох категорій для створення відповідних сегментів:

- «гарячі» дані, що оновлюються або вичитуються дуже часто та потребують доступу з низькими затримками;
- «теплі» дані, що використовуються менш часто, але все-одно потребують регулярного доступу;
- «холодні» дані, що рідко використовуються.

Після цього розподілена система може автоматично підібрати найбільш підходящі вузли для відповідного типу сегмента. Наприклад, сервери з найкращою швидкістю та найбільшою пропускною здатністю мережі повинні в першу чергу працювати з сегментами з «гарячими» даними, коли-як інші сегменти можуть обслуговуватись менш потужними вузлами.

Адаптивність в цьому методі забезпечується рядом додаткових технік. По-перше, алгоритм кластеризації передбачає постійний моніторинг активності в розподіленій системі та оновлення класифікації записів при зміні патернів доступу [16]. Крім того, зібрана статистика також дозволяє прогнозувати зміни в навантаженні кластера й предитивно змінювати конфігурацію сегментів або ж вузлів системи в цілому. З іншого боку, реалізація та використання такого функціоналу може вимагати застосування ШІ або машинного

навчання, що в свою чергу значно збільшує апаратні вимоги для розглянутого методу темпоральної сегментації даних.

Деякі інші методи сегментації намагаються розвивати та покращувати ідеї географічної сегментації. Наприклад, розробники сервісу Akkio пропонують концепцію мікро-шардів – значно зменшених за розміром сегментів даних, що формуються клієнтськими застосунками, а не самою розподіленою системою [17].

Зазвичай під час сегментації даних формують відносно великі (до декількох десятків гігабайт) шарди – цього достатньо для більшості інформаційних систем та спрощує проблему менеджменту сегментів при реплікації та інших задачах. Проте, в розподілених системах з географічно віддаленими вузлами вартість реплікації та міграції сегментів значно вища. При цьому вміст або структура окремого шарду може бути погано оптимізована для типових запитів на інших серверах системи.

Akkio пропонує зменшити розмір типового сегмента даних до декількох мегабайтів, а замість розподілу на основі діапазону або хешу – дозволити клієнтам розподіленої системи, наприклад СКБД, самостійно формувати ці мікро-шарди на основі даних, що часто використовуються тим чи іншим застосунком. Така техніка також дозволяє враховувати патерни доступу, як і вже розглянутий адаптивний метод сегментації, проте переносить частину функціоналу з самої розподіленої системи. З одного боку це дозволяє зменшити навантаження на вузли, але з точки зору безпеки все-одно необхідна реалізація додаткового механізму, що не дозволить клієнтам зловживати можливістю самостійно формувати шарди.

Основна перевага мікро-шардів, окрім адаптивності до патернів доступу, полягає в оптимізації реплікації. Копіювання невеликої кількості даних робить процес менш схильним до збоїв, а процес оновлення вмісту вузлів стає більш гнучким та швидким, що добре підходить для великих розподілених систем зі значною кількістю запитів [17].

З архітектурної точки зору, Akkio керує мікро-шардами за допомогою трьох основних компонентів: сервісу позиціонування, сервісу обліку доступу та сервісу розміщення даних. Сервіс позиціонування (англ. Location Service) зберігає інформацію про поточне розміщення сегментів, що необхідно для швидкого роутингу запитів між вузлами розподіленої системи.

Сервіс обліку доступу (англ. Access Counter Service) виступає в ролі системи моніторингу, що записує інформацію про мікро-шарди, що були використані для обробки запису, а також про саме джерело цих запитів. Сервіс розміщення даних (англ. Data Placement Service) може використовувати ці дані для виконання двох основних задач: формування і розміщення сегментів з новими даними, а також вибір нового місцезнаходження мікро-шардів під час міграції даних або балансування розподіленої системи. Остання операція також вимагає комунікації з сервісом обліку даних для підтримки актуальної інформації для майбутніх запитів.

Хоча процес міграції є швидким через мінімальний розмір сегментів, але для забезпечення узгодженості даних може бути необхідним застосування механізму блокувань або транзакцій – конкретний алгоритм дій залежить від розподіленої СКБД, з якою Akkio інтегрують.

В цілому, Akkio пропонує перспективний напрям розвитку класичної географічної сегментації даних, але має ряд обмежень і недоліків. По-перше, аналогічно до попереднього методу Akkio може вимагати значних апаратних ресурсів – його три складові сервіси формують з себе окрему систему, що потребує окремого сховища даних, а також високої швидкодії. Більшість операцій в Akkio є асинхронними, окрім локалізації сегменту через сервіс позиціонування, тому зайві затримки на цьому кроці можуть вповільнити всю розподілену систему. Сценарії застосування Akkio аналогічні до класичної географічної сегментації, тому для більш класичної високонавантаженої розподіленої системи інтеграція цього сервісу (або сервісу з аналогічним механізмом сегментування) може бути недоцільною.

Окрім системи, що була запропонована в Akkio, є й інші методи на основі географічної

сегментації, як-от сегментація з урахуванням мережі (англ. Network-Aware Sharding). Цей метод добре підходить для соціальних мереж та інших розподілених систем або СКБД, де в основі структури даних лежить велика кількість об'єктів або користувачів, що взаємодіють одне з одним.

Сегментація з урахування мережі пропонує алгоритм збору цих взаємодій і формування графу на їх основі [18]. Отриманий граф може бути використаний для формування фрагментів. Взаємопов'язані користувачі будуть розподілені в один і той же сегмент даних, що потенційно зменшить кількість міжсегментних транзакцій та покращить швидкодію розподіленої системи. Основним недоліком цього методу є його вузькоспеціалізованість – розподілені СКБД з іншими структурами даних, що наприклад не використовуються для запису взаємодії об'єктів, не зможуть досягти значного покращення швидкості виконання запитів.

Всі розглянуті методи передбачають сегментацію зберігання, але певного покращення швидкодії можна досягти й за допомогою використання лише сегментації обробки. Одне з досліджень пропонує використовувати сегментацію запитів на основі знань (англ. Knowledge-Driven Query Sharding).

Цей метод пропонує використання окремого алгоритму для розділення комплексних запитів, що можуть потребувати даних з різних джерел, таблиць чи документів, на менші підзапити. Їх, в свою чергу, вже можна виконувати паралельно для пришвидшення виконання [19]. В розподілених СКБД це може дати можливість розділити комплексні міжсегментні транзакції на більш прості, що будуть виконуватись окремо на вузлах з більш підходящим для роботи контекстом.

Проте, так як виконується лише сегментація обробки, цей метод можна застосувати й для покращення швидкодії класичних реляційних СКБД без підтримки розподіленого режиму обробки. Розглянутий метод може бути виконаний в якості окремого сервісу, що працює окремо від процесору запитів інформаційної системи, подібно до реалізації Akkio.

Основна відмінність цієї сегментації запитів від простого розпаралелювання запитів в інших СКБД – це вміння ділити запити на основі знань про структуру даних в системі. Це дозволяє чітко виділити окремі сегменти, де підзапит не буде потребувати доступу до інших записів. Відповідно, сервіс сегментації розділяє початковий запит чи транзакцію на ці незалежні підзапити, після чого вони передаються вже в саму розподілену систему, де процесор запитів може зробити власні додаткові оптимізації. Але в цьому також полягає і основне обмеження сегментації запитів на основі знань – реалізація методу для СКБД з попередньо відомою структурою даних може бути тривіальною, але розробка сервісу з динамічно змінюваною структурою може потребувати застосування ШІ, машинного навчання або інших підходів. Крім того, в оригінальному дослідженні також згадується проблема неефективності роботи запропонованого методу на малих запитах.

В цілому, можна виділити наступні загальні переваги різних адаптивних підходів, що пропонуються в розглянутих методах:

- можливості забезпечення динамічного перебалансування сегментів без втрати узгодженості;
- можливості поєднання вузлів з різними апаратними характеристиками без потреби вручну визначати розподіл даних між ними;
- потенційному зменшенні часу виконання міжсегментних транзакцій;
- можливості інтеграції з алгоритмами машинного навчання або іншими засобами ШІ для автоматичного визначення типових шаблонів доступу;
- потенційному покращенні використання обчислювальних та мережевих ресурсів.

Розглянуті особливості роботи методів сегментації даних можна об'єднати разом для більш детального аналізу функціональних характеристик, що наведено в таблиці 1. Було проаналізовано і класичні (статичні), і більш нові адаптивні методи. Розглянемо більш

детально обрані характеристики та методику оцінювання.

Таблиця 1

Порівняння методів сегментації даних

Метод	Динамічний розподіл даних	Адаптивність до змін	Складність реалізації	Апаратні вимоги	Локалізація даних
Діапазонна сегментація	-	-	Низька	Низькі	-
Хеш-сегментація	+/-	-	Низька	Низькі	-
Географічна сегментація	+/-	+/-	Низька/середня	Низькі/середні	+
Темпоральна сегментація	+	+	Висока	Високі	-
Мікро-шарди (Akkio)	+	+	Висока	Високі	+
Сегментація з урахуванням мережі	+	+	Середня	Середні	+
Сегментація запитів на основі знань	-	+/-	Низька/середня	Низькі/середні	-

По-перше, методи можна швидко порівняти за їх властивостями динамічно розподіляти дані між сегментами – ця характеристика враховує вміння методу використовувати додаткову інформацію для прийняття результуючого рішення. Відповідно, діапазонна сегментація використовує дуже простий статичний розподіл, де кожен шард є частиною певного діапазону значень. Хеш-сегментація та географічна сегментація дозволяють отримати кращий розподіл, що зменшує ймовірність отримати «гарячий» сегмент, який буде відповідальний за більшість запитів, тому вони отримують середню оцінку за цю характеристику. Найкраще себе показують з точки зору розподілу методи темпоральної сегментації, мікро-шардів та сегментації з урахуванням мережі. Сегментація запитів на основі знань не передбачає змін в системі зберігання, тому відповідно не надає жодного динамічного розподілу.

Адаптивність до змін характеризує можливість методу підлаштовуватись до змін в патернах доступу до даних під час роботи. Більш класичні методи не мають такої можливості, єдиним виключенням є географічна сегментація, що повинна враховувати місцезнаходження клієнтів розподіленої системи. Хоча, інші додаткові динамічні механізми цей метод не визначає, тому кінцеві можливості залежать від конкретної реалізації. Інші методи, за виключенням сегментації на основі знань, де така можливість може бути присутня в залежності від реалізації, не повинні мати проблем з підлаштуванням наявної розподіленої системи до нових патернів доступу. Сегментація запитів на основі знань напряду передбачає роботу лише з попередньо відомою структурою даних, хоча потенційно цей метод можна розширити до більш динамічного варіанту.

Технічні апаратні вимоги та складність реалізації дуже пов'язані між собою для методів сегментації даних. Всі класичні методи мають низьку або середню складність та вимоги. Темпоральна сегментація та мікро-шарди мають найбільші вимоги як до процесу розробки, так і для виконання через комплексні системи моніторингу запитів та інших додаткових характеристик системи. Сегментація з урахуванням мережі та на основі знань є більш простими методами, чиї вимоги більше залежать від розміру розподіленої системи та структури даних, що використовується в них.

Остання характеристика, локалізація даних, відповідає за вміння враховувати місцезнаходження клієнтів системи для більш ефективного розміщення шардів. Відповідно, з цією задачею можуть добре впоратись методи географічної сегментації, мікро-шардів/Akkio та сегментації з урахуванням мережі.

Так як розглянуті методи сегментації даних не мають еталонної реалізації для об'єктивного порівняння в однакових умовах функціонування, то зіставлення їх кількісних метрик, як-от мережових затримок, пропускну здатності, витрат на реплікацію сегментів

тощо, не є можливим. Наявні дослідження покладаються саме на ці метрики, що дозволяє отримати більш об'єктивне представлення реальної поведінки інформаційної системи з певним методом [20]. Проте, такий аналіз вимагає уважного підбору коректних варіантів реалізації та комплексного контролю за середовищем виконання, що значно збільшує час тестування та отримання кінцевих результатів [21]. Тому, запропоновано новий інтегральний критерій ефективності E на основі отриманих даних з таблиці 1. Критерій розраховується таким чином:

$$E = \frac{D + A + L}{1 + \beta H}, \quad (1)$$

де D – оцінка динамічного розподілу даних в методі сегментації даних, A – оцінка адаптивності методу до змін, L – оцінка локалізації даних, а H та β – оцінка та коефіцієнт апаратних вимог для застосування методу.

Відповідно, перші три оцінки мають адитивний вплив через їх незалежний вплив на методи сегментації – для вирішення певних вузькоспеціалізованих задач може бути достатньо лише однієї з розглянутих характеристик. При цьому, апаратні вимоги є обернено пропорційними через потенційне ускладнення масштабування системи. Наприклад, в умовах обмежених ресурсів, як-от при кордонних обчисленнях (англ. Edge Computing), може бути неможливим застосування потужнішого, але більш енергоємного обладнання, що в свою чергу призведе до деградації швидкодії методів, що будуть використовувати різноманітні інтелектуальні системи або засоби ШІ для прийняття рішень.

Так як підвищені апаратні вимоги негативно впливають на інтегральний показник, введено коефіцієнт β , що дозволяє зберегти значимість H в загальних розрахунках, але уникнути переважаючого впливу над іншими показниками. Крім того, в розрахунках запропонованого критерію ефективності не враховується складність реалізації, так як вона оцінює лише додаткові ресурси на етапі розробки, що будуть необхідні для інтеграції методу в СКБД, але це напряму не впливає на працездатність та функціональність готового рішення.

Характеристики порівняння методів з таблиці 1 було трансформовано в числові оцінки для використання у запропонованій формулі (1). Оцінка апаратних вимог у форматі «низькі-високі» відповідає балам від 1 до 3, коли-як інші характеристики в форматі «+», «+/-» та «-» перетворено в 1, 0,5 та 0 балів відповідно. Отримані значення запропонованого інтегрального критерію ефективності є відносними й їх можливі діапазони сильно відрізняються в залежності від використовованого коефіцієнта ваги апаратних вимог. Нормалізацію критерію ефективності до діапазону $[0;1]$ можна виконати таким чином:

$$E_n = \frac{E - E_{\min}}{E_{\max} - E_{\min}}. \quad (2)$$

При $D = A = L = 0$ мінімальне можливе значення критерію ефективності $E_{\min}$ завжди буде 0 незалежно від значення знаменника:

$$E_{\min} = \frac{0 + 0 + 0}{1 + \beta H} = 0. \quad (3)$$

Максимальне можливе значення $E_{\max}$ буде досягатись за мінімальних апаратних вимогах ($H = 1$), проте воно залежне від коефіцієнта β :

$$E_{\max} = \frac{1 + 1 + 1}{1 + \beta \cdot 1} = \frac{3}{1 + \beta}. \quad (4)$$

Базуючись на цьому, формула (2) може бути приведена до нормалізованого вигляду таким чином:

$$E_n = \frac{E - E_{\min}}{E_{\max} - E_{\min}} = \frac{E}{E_{\max}} = \frac{(D + A + L)(1 + \beta)}{3(1 + \beta H)}. \quad (5)$$

Для розрахунків емпірично обрано коефіцієнт ваги $\beta = 0,5$, що дозволяє забезпечити збалансований вплив апаратних вимог на загальну оцінку методів сегментації даних. Отримані оцінки критерію ефективності в звичайному та нормалізованому форматі для всіх розглянутих методів наведено в таблиці 2.

Таблиця 2

Інтегральний критерій ефективності методів сегментації даних

Метод	Динамічний розподіл даних, D	Адаптивність до змін, A	Локалізація даних, L	Апаратні вимоги, H	Критерій ефективності, E	Нормалізований критерій ефективності, E_n
Діапазонна сегментація	0	0	0	1	0	0
Хеш-сегментація	0,5	0	0	1	0,33	0,16
Географічна сегментація	0,5	0,5	1	2	1	0,5
Темпоральна сегментація	1	1	0	3	0,8	0,4
Мікро-шарди (Akkio)	1	1	1	3	1,2	0,6
Сегментація з урахуванням мережі	1	1	1	2	1,5	0,75
Сегментація запитів на основі знань	0	0,5	0	2	0,25	0,125

Коефіцієнт β може бути використаний для моделювання різних умов застосування для вибору найбільш підходящого методу сегментації. Відповідно, значення 0 повністю ліквідує вплив апаратних вимог в розрахунках – такий сценарій може використовуватись в хмарних середовищах з умовно необмеженими ресурсами. Значення >1 може бути використано для підбору методу сегментації для застосування в обмежених умовах – IoT системах, портативних девайсах тощо. Використання значення ≤ 1 в таблиці 1 дозволяє уникнути ситуації, коли більш функціональний метод має менший інтегральний показник через підвищені апаратні вимоги. Для дослідження впливу значення коефіцієнта β на критерій ефективності було проведено додаткові розрахунки, результати візуалізовано на рис. 2.

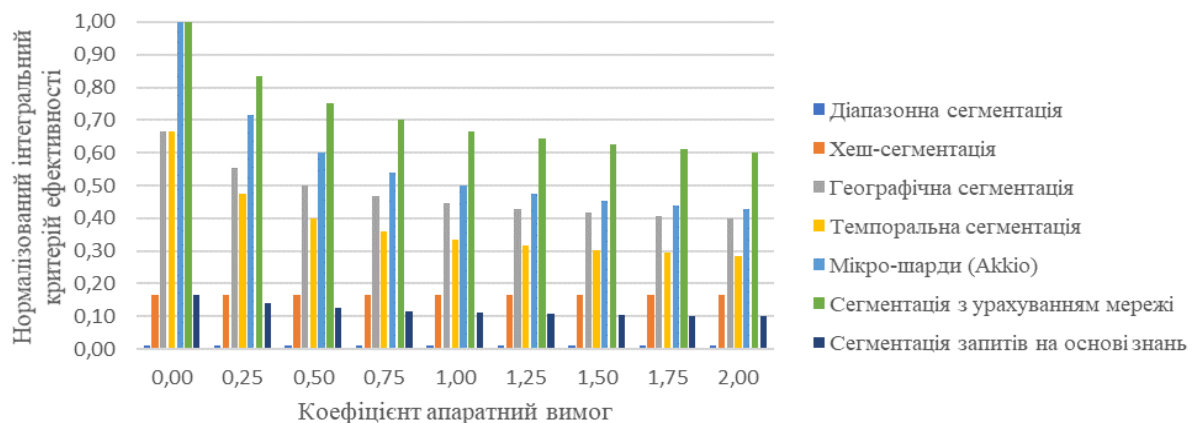


Рис. 2. Вплив коефіцієнту апаратних вимог на інтегральний критерій ефективності

На діаграмі можна прослідкувати декілька тенденцій. По-перше, ефективність методів сегментації даних зменшується зі зростанням коефіцієнта ваги апаратних вимог та має гіперболічний, а не лінійний характер, що пояснюється самим методом розрахунку нормалізованого критерію. Проте, діапазонна сегментація, хеш-сегментація та сегментація на основі знань мають стабільно низький інтегральний показник, що майже не змінюється. Це пояснюється їх низькою функціональністю в обраних для оцінювання характеристиках. Варто відмітити, що це не робить методи неефективними в загальному розумінні, а лише підкреслює їх обмежені можливості в сучасних інформаційних системах, що можуть вимагати адаптивності до патернів доступу, перерозподілу сегментів тощо.

Сегментація з урахуванням мережі зберігає стабільно високий інтегральний критерій ефективності, що пояснюється найкращою функціональністю серед обраних характеристик та помірними апаратними вимогами. При цьому, методи темпоральної сегментації та мікро-шардів показують гіршу тенденцію через вищі показники вимог. Особливо помітно це за високих значеннях коефіцієнта, де інтегральний критерій ефективності географічної сегментації майже досягає значень інших адаптивних методів. На основі цього можна зробити висновок, що в сценаріях застосування з обмеженими ресурсами варто розглядати й більш прості методи сегментації з обмеженими засобами адаптивності.

Запропонований інтегральний критерій ефективності є гнучким інструментом порівняння методів сегментації, який поєднує декілька критеріїв оцінювання, адаптивність до різних сценаріїв застосування і можливість нормалізації, а також формалізує раніше якісні характеристики. Це дозволяє кількісно оцінювати як класичні, так і більш сучасні динамічні або адаптивні методи, роблячи модель корисною для досліджень і практичного проектування розподілених СКБД. Більш того, запропоновані для оцінювання характеристики можна змінювати на інші або доповнювати в майбутньому для формування більш об'єктивних та усесторонніх засобів оцінювання методів сегментації.

Таким чином, більшість розглянутих методів показують покращення і розвиток сегментації даних в СКБД та розподілених системах в цілому. Динамічна або адаптивна сегментація даних, що враховує характер та структуру даних, дозволяє покращити швидкодію системи. Хоча розглянуті методи є більш вузькоспеціалізованими, але їх основні ідеї можна використати для нового методу сегментації даних з більш універсальним підходом до покращення швидкодії обробки запитів та реплікації даних.

Висновки

У роботі проведено систематизований аналіз класичних та динамічних методів сегментації даних у розподілених системах, що дозволило виявити їх функціональні відмінності та обмеження. На основі формалізації ключових характеристик – динамічності розподілу, адаптивності, локалізації даних та апаратних вимог – запропоновано інтегральний критерій ефективності, який забезпечує кількісне порівняння методів сегментації незалежно від їхньої архітектури та конкретної реалізації.

Розраховані значення критерію ефективності показали, що методи темпоральної сегментації та мікро-шардів демонструють найвищу ефективність у сценаріях з достатніми обчислювальними ресурсами, що пояснюється їх здатністю адаптуватися до зміни патернів доступу. Водночас сегментація з урахуванням мережевої структури виявилася більш стабільною у ресурсно обмежених середовищах, забезпечуючи високий рівень локалізації даних при помірних вимогах до обладнання. Класичні статичні методи отримали низькі оцінки інтегрального критерію через відсутність адаптивності, однак залишаються доцільними для систем із передбачуваними навантаженнями та простими патернами доступу.

Дослідження залежності критерія ефективності від вагового коефіцієнта апаратних вимог засвідчило, що вибір оптимального методу сегментації визначається не лише його

алгоритмічними властивостями, а й умовами застосування – типом інфраструктури, динамікою навантаження та вимогами до продуктивності. Це підтверджує доцільність використання запропонованого критерію як інженерного інструменту для обґрунтованого вибору техніки сегментації під конкретний сценарій.

Практична цінність отриманих результатів полягає в можливості застосування інтегрального критерію для попереднього моделювання та оптимізації архітектури розподілених СКБД без необхідності створення експериментальних прототипів кожного методу. Перспективою подальших досліджень є розроблення адаптивних методів і алгоритмів сегментації, які враховуватимуть не лише патерни доступу, але й моделі узгодженості даних, а також інтеграція методів машинного навчання для автоматичного визначення оптимального розподілу сегментів у реальному часі.

СПИСОК ЛІТЕРАТУРИ

1. Zulkifli A. Accelerating database efficiency in complex IT infrastructures: advanced techniques for optimizing performance, scalability, and data management in distributed systems. *International journal of information and cybersecurity*. 2023. Vol. 7, №12. P. 81–100.
2. Ayyagiri A., Pandian P. K. G., Goel P. Efficient data migration strategies in sharded databases. *Journal of quantum science and technology*. 2024. Vol. 1, №2. P. 72–87. URL: <https://doi.org/10.36676/jqst.v1.i2.17> (date of access: 04.10.2025). Title from screen.
3. Solat S. Sharding distributed databases: a critical review. *ArXiv preprint*. 2024. URL: <https://doi.org/10.48550/arXiv.2404.04384>. Title from screen.
4. DYNASHARD: secure and adaptive blockchain sharding protocol with hybrid consensus and dynamic shard management / Ao Liu et al. *IEEE internet of things journal*. 2024. URL: <https://doi.org/10.1109/jiot.2024.3490036> (date of access: 05.10.2025). Title from screen.
5. Dritsas E., Trigka M. A survey on database systems in the big data era: architectures, performance, and open challenges. *IEEE access*. 2025. URL: <https://doi.org/10.1109/access.2025.3572059> (date of access: 05.10.2025). Title from screen.
6. Solat S., Nait-Abdesselam F. Sharding distributed replication systems to improve scalability and throughput. *Building cybersecurity applications with blockchain and smart contracts*. Cham, 2023. P. 101–124. URL: https://doi.org/10.1007/978-3-031-50733-5_5 (date of access: 05.10.2025). Title from screen.
7. Research on transaction allocation strategy in blockchain state sharding / Guangxia Xu et al. *Future generation computer systems*. 2025. URL: <https://doi.org/10.1016/j.future.2025.107756> (date of access: 05.10.2025). Title from screen.
8. Aggregate based light incremental sharding for efficient embedding table management for recommender systems / Chao Kong et al. *Scientific reports*. 2025. Vol. 15, № 1. URL: <https://doi.org/10.1038/s41598-025-15351-8> (date of access: 05.10.2025). Title from screen.
9. Sharding by hash partitioning - A database scalability pattern to achieve evenly sharded database clusters / Caio H. Costa et al. *17th international conference on enterprise information systems*, Barcelona, Spain, 27–30 April 2015. [S. l.], 2015. URL: <https://doi.org/10.5220/0005376203130320> (date of access: 05.10.2025). Title from screen.
10. Mao C., Golab W. GeoChain: a locality-based sharding protocol for permissioned blockchains. *ICDCN 2023: 24th International conference on distributed computing and networking*, Kharagpur India. New York, NY, USA, 2023. URL: <https://doi.org/10.1145/3571306.3571392> (date of access: 05.10.2025). Title from screen.
11. Shard manager / Sangmin Lee et al. *SOSP '21: ACM SIGOPS 28th symposium on operating systems principles, Virtual Event Germany*. New York, NY, USA, 2021. URL: <https://doi.org/10.1145/3477132.3483546> (date of access: 05.10.2025). Title from screen.
12. Петух А. М., Романюк О. В., Романюк О. Н. Базы даних. Мови запитів, управління транзакціями, розподілена обробка даних. Вінниця : ВНТУ, 2016. 95 с.

13. Goli M., Rankoohi S. M. T. R. A new vertical fragmentation algorithm based on ant collective behavior in distributed database systems. *Knowledge and information systems*. 2011. Vol. 30, № 2. P. 435–455. URL: <https://doi.org/10.1007/s10115-011-0384-6> (date of access: 05.10.2025). Title from screen.
14. Миргородський А. В., Романюк О. В. Аналіз архітектурних принципів сучасних розподілених систем керування базами даних. *Матеріали LIV Всеукраїнської науково-технічної конференції підрозділів ВНТУ*, Вінниця, 24–27 берез. 2025 р. Вінниця, 2025. URL: <http://ir.lib.vntu.edu.ua/handle/123456789/48567?show=full>.
15. Миргородський А. В., Романюк О. В., Романюк О. Н. Аналіз динамічних моделей забезпечення узгодженості даних у розподілених системах керування базами даних. *Measuring and computing devices in technological processes*. 2025. № 3. С. 285–292. URL: <https://doi.org/10.31891/2219-9365-2025-83-35>. Назва з екрана.
16. Self-healing Nodes with Adaptive Data-Sharding / Ayush Thakur et al. *ArXiv preprint*. 2024. URL: <https://doi.org/10.48550/arXiv.2405.00004>. Title from screen.
17. Sharding the shards: managing datastore locality at scale with Akkio / Muthukaruppan Annamalai et al. *Proceedings of the 13th USENIX conference on operating systems design and implementation*. USA, 2018. P. 445-460.
18. Sharding social networks / Quang Duong et al. *The sixth ACM international conference*, Rome, Italy, 4–8 February 2013. New York, New York, USA, 2013. URL: <https://doi.org/10.1145/2433396.2433424> (date of access: 05.10.2025). Title from screen.
19. Krasuski A., Szczuka M. Knowledge driven query sharding. *Informatik-Berichte*. 2012. Vol. 225. P. 182–190.
20. Jayaraman S., Borada D. Efficient data sharding techniques for high-scalability applications. *Integrated journal for research in arts and humanities*. 2024. Vol. 4, №6. P. 323–351. URL: <https://doi.org/10.55544/ijrah.4.6.25> (date of access: 13.11.2025). Title from screen.
21. Salunke S. V., Ouda A. A performance benchmark for the PostgreSQL and MySQL databases. *Future internet*. 2024. Vol. 16, №10. P. 382. URL: <https://doi.org/10.3390/fi16100382> (date of access: 13.11.2025). Title from screen.

Стаття надійшла до редакції 20.10.2025.

Стаття пройшла рецензування 20.11.2025.

Миргородський Андрій Вікторович – аспірант кафедри програмного забезпечення, e-mail: mirgorodskijav@gmail.com.

Романюк Оксана Володимирівна – канд. техн. наук, доцент кафедри програмного забезпечення, e-mail: romaniukoksana@gmail.com.

Вінницький національний технічний університет.