

ТЕХНІЧНІ НАУКИ

УДК 004.652.4

DOI: 10.15587/2313-8416.2017.108420

АНАЛІЗ ЧАСТКОВОГО ВИРІШЕННЯ ПРОБЛЕМИ ТЕОРЕМИ БРЮЄРА В СХОВИЩАХ ДАНИХ В ОПЕРАТИВНІЙ ПАМ'ЯТІ

© О. Ю. Добровольський

Key-value сховища даних в оперативній пам'яті об'єднують в собі підходи BASE-систем і алгоритми оновлення індексів, що розраховувались з метою роботи в глобальній мережі. Теорема Брюєра, під яку підпадають такі сховища, стверджує що не можливо забезпечити одразу три головні властивості таких систем. Для підвищення консистентності без шкоди для горизонтального масштабування системи пропонується використання синхронних індексів

Ключові слова: теорема Брюєра, IMDG, консистентність, індекс, доступність, розподіленість, key-value, оперативна пам'ять

1. Вступ

Реляційні бази даних зобов'язані задовольняти вимоги ACID [1] (atomicity, consistency, isolation, durability), тобто ключовим поняттям в таких системах є транзакція, як група об'єднаних послідовних операцій з базою даних. Транзакція не може бути виконана частково, не порушує узгодженість системи, не впливає на одночасну роботу інших транзакцій [2]. Крім того, виконані успішною транзакцією зміни не повинні бути втрачені внаслідок будь-яких збоїв системи (наприклад, знеструмлення або збою в обладнанні). Ці вимоги добре відповідають традиційним бізнес-моделям, проте повною мірою реалізувати їх в розподілених системах, з урахуванням необхідної швидкодії, вкрай складно.

2. Літературний огляд

Теорема Брюєра, стверджує, що розподілена система здатна забезпечити не більше двох з трьох властивостей: консистентність, доступність, стійкість до розподілу (Consistency, availability, partition tolerance) [3]. Кожна з цих властивостей є важливою для сучасних систем [4], тому відбуваються пошуки можливих рішень теореми і забезпечення віх трьох властивостей в повному обсязі [5].

Сучасні сховища типу ключ-значення спроектовані з урахуванням теореми Брюєра [6]. За сучасною кваліфікацією такі сховища відносяться до BASE-систем [7]. Вони забезпечують «базову доступність» і «узгодженість в кінцевому рахунку» даних, що зберігаються [8]. Це дозволяє забезпечити високу потенційну масштабованість і здатність системи до ефективного функціонування при великих відстанях між вузлами. Однак значна частина реальних випадків використання таких сховищ демонструє розгор-

тання системи в рамках однієї високошвидкісної локальної мережі передачі даних. В таких умовах штатні механізми індексування розподілених сховищ, розраховані на роботу в глобальній мережі з великими затримками, стають неефективними і призводять до надмірних затримок в оновленні індексів, а, отже, до втрати консистентності між індексами і індексованими даними.

3. Мета та задачі дослідження

Мета дослідження – розробка методу підвищення консистентності і доступності даних, що містяться в розподілених сховищах даних в оперативній пам'яті типу «ключ-значення» без втрати стійкості таких сховищ до поділу.

Для досягнення мети були поставлені наступні задачі:

1. Описати метод синхронного індексування.
2. Реалізувати модель для тестування методу.

4. Вирішення проблеми консистентності за рахунок синхронних індексів

4.1. Опис методики синхронного індексування

У дослідженні для підвищення консистентності і доступності даних в розподілених сховищах типу «ключ-значення» пропонується ввести синхронні індекси за вторинними полями. Вибір структури такого індексу повинен, з одного боку, максимально відповідати ідеології зберігання даних, а з іншого – забезпечити швидкий і сумісний результат призначеного для користувача запиту. При цьому індекс повинен задовольняти наступні вимоги:

- здатність забезпечити швидкий спеціальний пошук по значенням необхідних параметрів;
- високою швидкістю оновлення індексу;

- безперервної узгодженістю індексу і збережених даних;
- мати якомога менші накладні витрати.

Базовою структурою даних в документ-орієнтованих сховищах є розподілена хеш-таблиця [9]. Така концепція підходить і для зберігання індексу, тому що він ставить у відповідність одному ключу безліч значень. Однак, в разі, коли одне і те ж значення, що індексується поля має безліч документів, запис в індексі буде занадто великий. В цьому випадку для полегшення операцій пошуку і відновлення окремих ділянок індексу рекомендується сегментувати індекс, тобто розбити кожен запис на сегменти таким чином, щоб розмір кожного сегмента був невеликим.

У загальному вигляді структура сегмента індексу представлена на рис. 1. Такий сегмент зберігається як звичайний документ, що підвищує універсальність підходу і легкість його впровадження.

Сегментувати індекси можна за різними принципами. У разі, коли одному значенню індексованого поля відповідає дуже багато документів, запис розбивається на кілька частин. В разі ж, коли одному значенню відповідає мало документів, записи можна об'єднувати, розробивши правило, що дозволяє по необхідному значенню поля отримати ідентифікатор індексного запису. На рис. 2 представлений приклад такого індексу з об'єднаними записами, що представляє програму телепередач.

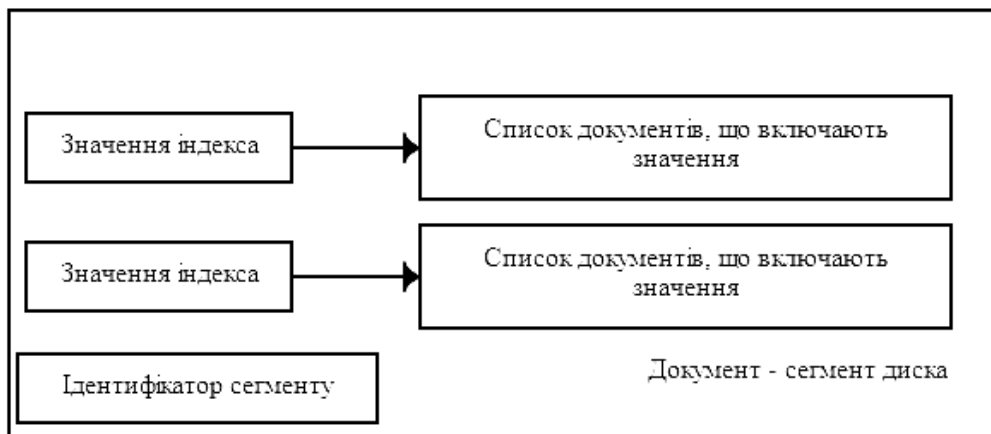


Рис. 1. Загальний вид сегмента синхронного індексу

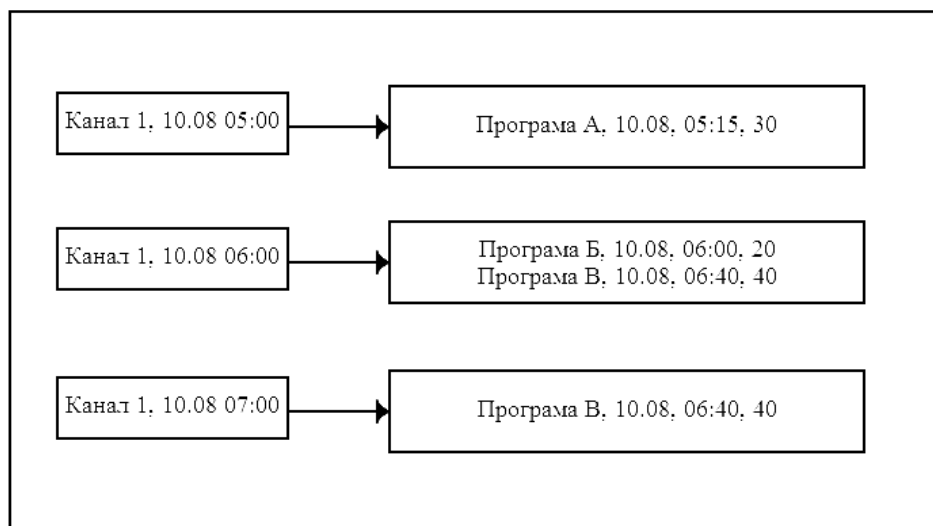


Рис. 2. Приклад синхронного індексу з об'єднанням записів в сегменті

У цьому індексі передбачаються користувацькі запити виду «отримати всі телепередачі, активні в заданий проміжок часу на даному каналі». Для обробки таких запитів використаний складений ключ, що включає ідентифікатор каналу і час. Оскільки в один момент часу на заданому каналі може бути активна тільки одна програма, для економії пропонується використовувати один сегмент для певного інтервалу часу (в прикладі – години), протягом якого існують активні телепередачі.

Тоді, крім ідентифікатора програми, необхідно також зберігати дійсний час її початку і тривалість, щоб точно знати, активна вона в заданий період часу всередині заданого проміжку.

При такому підході одна і та ж програма може зустрічатися в декількох індексних записах, якщо вона активна відразу в декількох проміжках часу. З іншого сторони, запропонована структура зручна при пошуку даних для діапазону індексних значень: знаючи границі часового проміжку, можемо відразу обчислити іденти-

фікатори сегментів, використовуючи потім найшвидший метод пошуку по ключу. У разі, коли індексовані значення не володіють вираженою періодичністю, але потрібно забезпечити пошук по діапазону значень, рекомендується використовувати деревовидні структури, в вузлах що містять ідентифікатори сегментів (рис. 3). При цьому не рекомендується використовувати В + дерева, як це робиться в реляційних базах даних. Як показано в [10], В + не забезпечує ефективну роботу системи управління базою даних в оперативній пам'яті, тому в даній роботі використані червоно-чорні дерева.

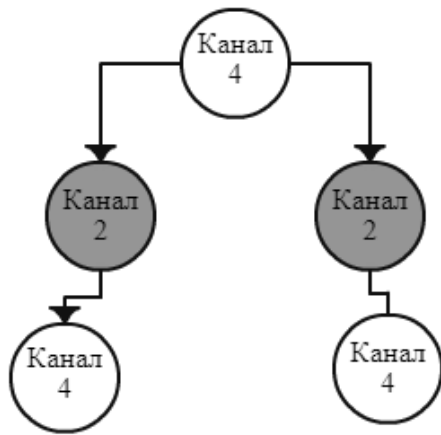


Рис. 3. Деревоподібна структура для пошуку діапазонна індексних значень

4.2. Алгоритм поновлення індексу

Для оновлення індексів при додаванні нових або оновлення існуючих документів пропонується виконувати наступний алгоритм: спочатку робиться спроба пошуку по ключу старої версії даного документа, вже записаною в даному сховищі. Потім обчислюються індексні записи для старої і нової версії об'єкта. Під індексним записом тут розуміється елементарна структура, яка містить ім'я індексу, ідентифікатор сегмента, до якого відноситься даний запис і дані, які повинні бути записані в індекс (рис. 4).

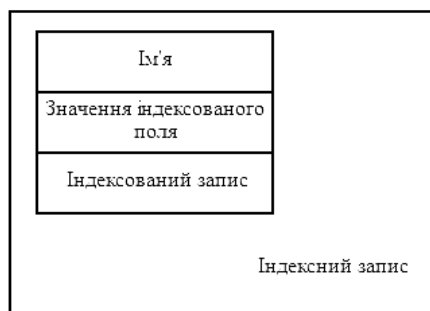


Рис. 4. Структура індексного запису

Виконується обчислення різниці множини старих і нових індексних записів (якщо об'єкта з даними ідентифікатором раніше не було в сховищі – множина старих індексних записів приймається порожньою). Потім для індексних записів, вже не присутніх в новому документі, виконується видалення, а для записів, новопосталих – додавання в сегмент.

5. Результати дослідження

Для відпрацювання методики синхронного індексування було вибрано розподілене документ-орієнтоване сховище Couchbase. Дане сховище демонструє високі швидкості читання / запису документів по ключу, але має досить недосконалим механізм виконання більш складних призначених для користувача запитів (рис. 5).



Рис. 5. Алгоритм оновлення індексу при додаванні документа

Для пошуку довільної інформації Couchbase пропонує концепцію видів (views), тобто функцій на JavaScript, за допомогою яких користувач сховища визначає зміст індексних записів. Процес оновлення видів запускається тільки після додавання 5000 но-

вих документів або по закінченню п'яти секунд після додавання документа, і займає, для великого сховища, до десяти хвилин. Під час оновлення видів запити до них відпрацьовують, повертаючи неактуальні (застарілі) дані. При цьому пошук документа по ключу завжди вибирає актуальну версію документа. Таким чином, враховуючи частоту операцій додавання документів, Couchbase може служити яскравим прикладом узгодженості в кінцевому рахунку, демонструючи відсутність про відсутність консистентності між видами і зображеними ними документами.

Наведемо приклад такої ситуації. Припустимо, вид відображає документи по вмісту їх поля «колір». Якийсь документ з ідентифікатором ID42 містив в цьому полі значення «зелений», що знайшло своє відображення відповідним видом, в якому значенням «зелений» був зіставлений ідентифікатор ID42. Потім документ з ID42 був оновлений, і в новій версії поле «колір» змінило значення на «синій», а, через деякий період, почалося оновлення виду.

Якщо користувач в період з моменту оновлення документа до закінчення процесу оновлення виду спробує, скориставшись вбудованими механізмом, знайти документ з кольором «зелений», він отримає

ідентифікатор ID42. При цьому, пошук документом та по ID42 поверне оновлений документ з кольором «Синій». Таким чином, часті операції оновлення призводять сховище в не консистентний стан, унеможливаючи пошук по чому-небудь, крім ідентифікаторів документів.

В експерименті використовувалися три індекси різних типів: індекс точної відповідності значення поля (індекс А), індекс відповідності значенню часових діапазонів (індекс Б), і індекс значень з можливістю пошуку по діапазону (індекс В).

Робота з синхронними індексами була реалізована за допомогою клієнтської бібліотеки Couchbase Java SDK 2.1 на клієнтській стороні. Couchbase сервер складався з двох вузлів, розміщених в одній локальній мережі з клієнтом.

Методика експерименту передбачала додавання в кожен індекс півмільйона записів, потім 10 тисяч операцій вибірок, потім додавання ще півмільйона записів (перекриття ідентифікаторів 25 %) з безперервним виконанням десяти тисяч запитів.

Поведінка кожного індексу порівнювався з поведінкою відповідного виду (view), що виконував ті ж завдання. Результати експерименту наведені в табл. 1.

Таблиця 1

Результати вимірів часу виконання і неконсистентності результатів виконання запитів

Індекси / види	Додавання 0,5 10 ⁶ записів, с	Виконання 10 ⁴ запитів, с	Оновлення індекса і виконання запитів, с	Кількість неконсистентних результатів, с
Індекс А	90	100	150	0
Індекс В	95	107	174	0
Індекс В	184	115	336	0
Індекс А,Б,В	187	110	283	0
Вид А	61	94	324	110
Вид Б	61	115	321	143
Вид В	62	110	335	214
Вид А,Б,В	63	107	542	178

6. Обговорення результатів дослідження

Як видно, час запису документів у порожнє сховище при використанні механізму синхронного індексування в 1,5–2 рази вище, ніж час аналогічною операції з використанням видів. Це пояснюється тим, що асинхронна перебудова індексів запускається трохи пізніше додавання, забезпечуючи менше навантаження на апаратне забезпечення. Найповільніше йде додавання в діапазонні індекси, що вимагають три операції запису і дві операції читання для запису одного об'єкта.

Виконання запитів на вільному від інших задач сховищі виконується приблизно однаково, як за допомогою view, так і за допомогою синхронних індексів з невеликим випередженням перших.

Нарешті, під час одночасного завантаження і читання даних асинхронні індекси показують

уповільнення продуктивності блокувань при оновленні.

7. Висновки

1. Метод синхронного індексування розподілених документ-орієнтованих сховищ даних показав свою придатність за умови, що клієнт і сервер знаходяться в одній локальній мережі.

2. Запропонований метод дозволив домогтися підвищення швидкості обробки запитів, і, одночасно, консистентності їх результатів при одночасній загрузці даних в сховищі. Синхронний індекс, що дозволяє запити по діапазонами, показав зменшення швидкості обробки запиту у порівнянні з асинхронними індексами, тому такий вид індексу рекомендується використовувати тільки в тому випадку, коли узгодженість індексів має першорядне значення.

Література

1. Cattell, R. Scalable SQL and NoSQL data stores [Text] / R. Cattell // ACM SIGMOD Record. – 2011. – Vol. 39, Issue 4. – P. 12–27. doi: 10.1145/1978915.1978919

2. Gray, J. The Transaction Concept: Virtues and Limitations [Text] / J. Gray // Proceedings of the 7th International Conference on Very Large Databases. – 1981. – P. 144–154.
3. Brewer, E. CAP twelve years later: How the “rules” have changed [Text] / Computer. – 2012. – Vol. 45, Issue 2. – P. 23–29. doi: 10.1109/mc.2012.37
4. Gilbert, S. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services [Text] / S. Gilbert, N. Lynch // ACM SIGACT News. – 2002. – Vol. 33, Issue 2. – P. 51–59. doi: 10.1145/564585.564601
5. Birman, K. Overcoming CAP with Consistent Soft-State Replication [Text] / K. Birman, D. Freedman, Q. Huang, P. Dowell // Computer. – 2012. – Vol. 45, Issue 2. – P. 50–58. doi: 10.1109/mc.2011.387
6. Gilbert, S. Perspectives on the CAP Theorem [Text] / S. Gilbert, N. Lynch // Computer. – 2012. – Vol. 45, Issue 2. – P. 30–36. doi: 10.1109/mc.2011.389
7. Pritchett, D. BASE: an ACID alternative [Text] / D. Pritchett // Queue. – 2008. – Vol. 6, Issue 3. – P. 48–55. doi: 10.1145/1394127.1394128
8. Bailis, P. Eventual consistency today [Text] / P. Bailis, A. Ghodsi // Communications of the ACM. – 2013. – Vol. 56, Issue 5. – P. 55–63. doi: 10.1145/2447976.2447992
9. Gupta, M. K. In-Memory Database Systems – A Paradigm Shift [Text] / M. K. Gupta, V. Verma, M. S. Verma // International Journal of Engineering Trends and Technology. – 2013. – P. 333–336.
10. Шапоренков, Д. А. Эффективные методы индексирования данных и выполнения запросов в системах управления базами данных в основной памяти [Текст]: дис. канд. физ.-мат. наук / Д. А. Шапоренков. – СПб., 2006. – 126 с.

*Рекомендовано до публікації д-р техн. наук Бузовський О. В.
Дата надходження рукопису 20.06.2017*

Добровольський Олександр Юрійович, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», пр. Перемоги, 37, м. Київ, Україна, 03056
E-mail: dobrunya777@gmail.com

УДК 536.62

DOI: 10.15587/2313-8416.2017.108935

АНАЛІЗ АПРОКСИМАЦІЇ РЕЗУЛЬТАТІВ ВИМІРЮВАНЬ ТЕПЛООВОГО ПОТОКУ БОМБОВОГО КАЛОРИМЕТРА В НЕСТАЦІОНАРНОМУ РЕЖИМІ

© **В. П. Бабак, А. О. Запорожець, О. О. Назаренко, О. О. Редько**

Досліджено принцип функціонування бомбового калориметра та методи визначення теплоти згорання палива. Проведені експериментальні дослідження по визначенню теплоти згорання дерев'яних пелетів. На основі проведених досліджень показані потенціальні зони скорочення часу встановлення показів результатів вимірювання бомбового калориметра. Запропоновано метод скорочення часу вимірювання теплоти згорання палива та проаналізовані його імовірнісні характеристики

Ключові слова: бомбовий калориметр, теплота згорання палива, інтерполяційні функції, коефіцієнт детермінації

1. Вступ

Забезпечення енергоефективності та енергозбереження є одним із пріоритетних напрямів розвитку сучасного інноваційно-технологічного суспільства. На підприємствах різних сфер промисловості для контролю енергетичної ефективності багатьох технічних засобів та оцінки теплотворної здатності палив в обов'язковому порядку використовуються калориметричні установки. Тому вирішення проблем метрологічного забезпечення вимірювань, що характеризуються теплотворну здатність енергетичних ресурсів, є одним із стратегічних завдань розвитку промисловості, зокрема паливно-енергетичного сектору. Перехід на новий техніко-економічний рівень обліку транспортування та постачання енергоресурсів за енергетичними показниками вимагає вирішення комплексу важливих науково-технічних проблем, орієнтованих на вдосконалення взаємопов'язаних еталонів, оснащення підприємств новітніми приладами та методами обліку енергетичних параметрів та досягнен-

ня сучасного рівня метрологічного забезпечення таких приладів.

2. Літературний огляд

До теперішнього часу на більшості підприємств для визначення теплотворної здатності різних видів енергетичних палив в основному використовуються бомбовий адиабатичний калориметр спалювання В2-08 [1, 2]. Випускаються також інші марки калориметрів, наприклад, АБК-1, що є аналогами В2-08. Вони базуються на вимірюванні зміни температури калориметричної рідини при спалюванні речовини в посуді («бомбі»), що розміщується в цій рідині [3]. Процес вимірювання є достатньо довгим (складає близько 30–40 хвилин) і в його основі лежать емпіричні формули алгоритму перетворення зміни температури в кількість теплоти. Процес вимірювання на цих калориметрах проходить в напівавтоматичному режимі і вимагає для роботи кваліфікованого оператора.