АНАЛІЗ ПРОДУКТИВНОСТІ СИСТЕМИ З ВИКОРИСТАННЯМ МЕТОДІВ ВИБОРУ СЕГМЕНТІВ В ELASTICSEARCH

Токар Л.О., Солоділов В.В., Гонтар І. Ю.

Кафедра інфокомунікаційної інженерії ім. В.В. Поповського,
Харківський національний університет радіоелектроніки,
Україна.

E-mail: liubov.tokar@nure.ua
iryna.hontar@nure.ua

Abstract

The paper analyzes the performance of the system using segment selection methods in Elasticsearch. Its use will allow to solve the issues of increasing scalability, consistency and fault tolerance of distributed databases and large search systems. It is shown that such an approach will make it possible to rank the documents that are most relevant to the user's query. The segment selection process is considered. It is shown that the selection of a fragment plays an important role in determining the relevant results for a query, and also affects the saving of resources and the increase of document search time. The paper analyzes the time delay spent by different sets of segments, and evaluates documents by different sets of segments for the tested queries.

Одним з методів ефективної роботи систем з паралельним виконанням операцій обробки даних по мережі є використання методу сегментування. Застосування даного методу дає можливість визначити, який конкретний сегмент або розділ необхідно шукати, щоб отримати відповідний документ, що відповідає запиту користувача, а також ранжувати документи, які мають найбільше відношення до запиту користувача. Сегментування – це метод поділу бази даних, який ділить дані по рядках й зберігає ці дані на кілька вузлів, які будуть працювати спільно паралельно для досягнення необхідної мети та підвищення продуктивності. Тому питання підвищення масштабованості, узгодженості та відмовостійкості мережі є дуже важливими, що й визначає актуальність роботи.

Метод сегментування використовується для підвищення продуктивності розподілених баз даних шляхом створення n-го числа сегментів або розділів. Сегмент або розділ – це частина бази даних, розподілена по різних вузлах. Сегментування забезпечує конфіденційність та безпеку. Очевидні основні переваги даного методу, які виражено в підвищеній продуктивності пошуку й меншому розмірі індексу, в надійності, в узгодженості і, як наслідок, в підвищенні швидкості обробки процесів [1].

Інструмент Elasticsearch є одним зі складових стеку програм ELK - Elasticsearch, Logstash і Kibana, що використовується для побудови рішень з моніторингу будь-якої інфраструктури. Elasticsearch є основою стека ELK та виступає як розподілений механізм пошуку та аналітики з відкритим кодом. Платформа Elasticsearch використовує техніку затінення й має високу масштабованість, відмовостійкість й забезпечує мінімальні втрати даних.

Elasticsearch – це пошуковий сервер, надає можливість роботи розподіленого в повному обсязі повнотекстового пошукового механізму в режимі реального часу. Однією із його ключових особливостей є можливість швидкого пошуку шляхом індексації тексту. Таким чином, Elasticsearch виграє в порівнянні з іншими пошуковими системами. Перевага Elasticsearch полягає в тому, що його можна запустити на своєму ноутбуці та масштабувати до сотень серверів й петабайт даних.

Ключові особливості Elasticsearch: забезпечує пошук й аналіз даних в режимі реального часу; є розподіленою системою, яка може працювати від скромного ноутбука до тисяч вузлів; може бути розгорнута для високодоступних кластерів із підтримкою багатонаціональності; після додавання нового вузла або відмови автоматично реорганізує й врівноважує дані; надає зручний інтерфейс

RESTful, використовуючи JSON через HTTP ток, що всі дані або інформація зберігаються як структуровані документи JSON.

Використання платформи Elasticsearch для налаштування виконання операції сегментування ґрунтується на таких фактах:

– колекція документів може бути розділена таким чином, щоб більшість відповідних документів для запиту зберігалася в декількох сегментах без попереднього знання набору запитів;

– коли дані розподілені за різними сегментами, ці сегменти можна ранжувати на основі їхньої релевантності запиту;

– можна оцінити мінімальну кількість верхніх сегментів у рейтингу, які потрібно знайти для запиту.

Процес вибору сегментів характеризує рис. 1 [1].

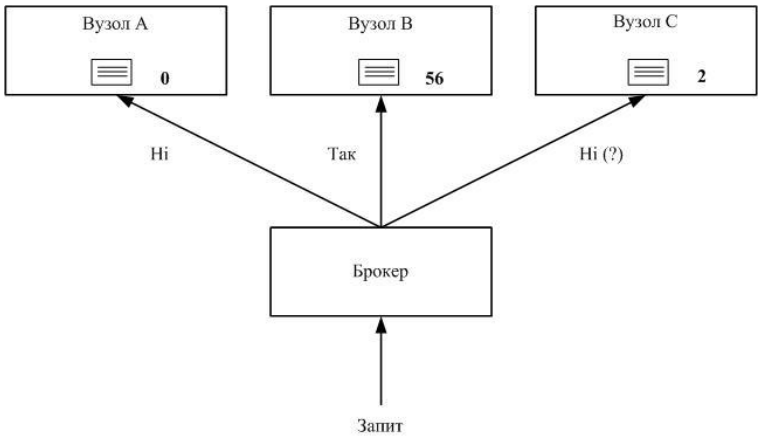

Рис. 1. Процес вибору сегментів

Користувач відправляє запит вузлу брокера, який буде перенаправляти запит на всі підключені вузли даних, тобто вузли A, B, C. Всі вузли даних можуть не містити відповідний документ, який відповідає запиту користувача. Тому в таких випадках кілька вузлів будуть вносити свій внесок в отримання найбільш важливих документів. Найбільш часто використовуваний підхід до вибору сегментів полягає в тому, щоб мати по одному сегменту в кожному вузлі даних.

Як тільки відповідні документи будуть витягнуті з вузла даних, вузол брокера об'єднає всі документи та поверне їх назад користувачеві. На рис. 1 більшість відповідних документів відносяться до вузла B, немає відповідних документів з вузлом A й дуже мало відповідних документів з вузлом C. Вузол брокера повинен ігнорувати пошук відповідних документів з вузлів A й C, але виникне ймовірність того, що вузол C може містити документи, які більш релевантні запиту в порівнянні з документами із вузла B. Тому вибір фрагмента відіграє важливу роль у визначенні релевантних результатів для даного запиту, а також зекономить ресурси й збільшить час пошуку документів.

Вихідні дані для аналізу обрано з урахуванням відомостей про них. Використовувалися три набори даних:

1) набір даних I представлено у форматі JSON й містить 10 000 документів за чотирма атрибутами;

2) набір даних II, що складається з 20 груп, який складається приблизно з 20 000 документів, розподілених (майже) рівномірно по 20 різним групам;

3) набір даних III –TREC. Це масив данних, який складає біля 75 000 екземплярів [2].

Таким чином, процес обчислення зводиться до знаходження подібності й ранжирування. Модель визначає оцінку подібності між запитом та документом. В цій моделі документ з найбільшою оцінкою частоти термінів вважається більш відповідним документу для даного запиту. Розгляд тільки частоти термінів може не знайти відповідності по відношенню до запиту користувача, тому частота термінів об'єднується зі зворотньою частотою документів для обчислення показника подібності [3].

Частота термінів та зворотня частота документів разом визначають показник подібності документа. Коли користувач надсилає запит, головний вузол аналізує ключове слово й знаходить оцінки подібності між документами. Потім документам присвоюються ранги на основі оцінки подібності. Першими витягуються документи з найвищим рейтингом.

Збір даних І розглядається з метою дослідження файлу у форматі JSON. Виконано операцію затінення з урахуванням точності при ранзі 10, 20 й 30. Тобто, наприклад, при k=5 означає, що результати запиту будуть знайдені в 5 кращих документах. У таблиці 1 показано оцінку точності в рангах 10, 20 й 30. У цьому випадку не застосовувався який-небудь конкретний алгоритм вибору сегментів, замість цього створені сегменти та проіндексовані документи у кожному з сегментів й виконано операцію пошуку.

Таблиця 1. Оцінка точності за рангами

Запит	k=10	k=20	k=30
1	33,3307	30,6089	29,0039
2	5,920	5,004	4,970
3	33,681	31,719	29,786

У таблиці 1 показано середнє значення оцінки точності, визначене для даного запиту. Виявлено, що чим більше оцінка точності, тим менша кількість відповідних документів витягується для даного запиту. У таблиці 2 інтерпретується процесорний час та оцінка подібності, які отримані для запиту в залежності від кількості сегментів.

Таблиця 2. Оцінка подібності в залежності від часу на пошук відповідного документа

Кількість сегментів	Час CPU, мс	Рахунок
1	42	6,5059
3	56	5,4694
6	80	4,8148

На рис. 2 представлено графіки затримки часу, витраченого різними наборами сегментів для тестованих запитів.

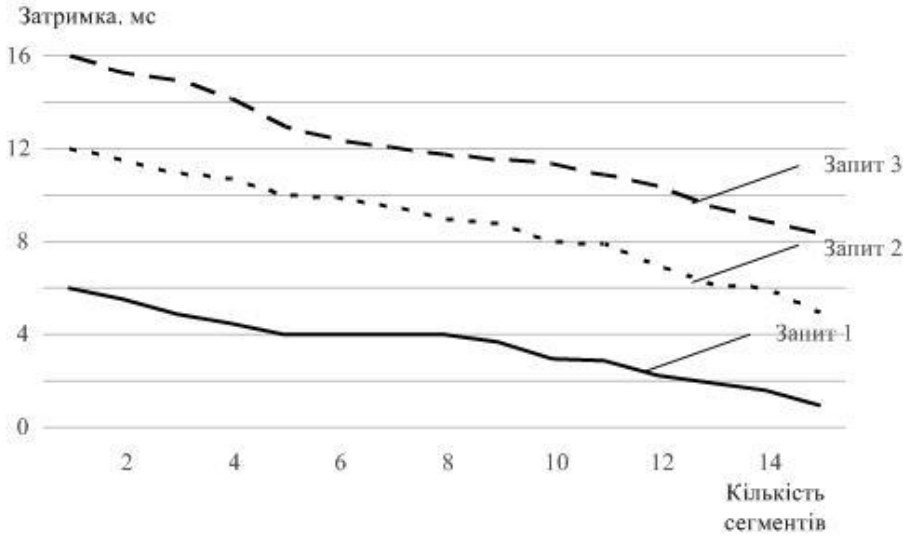

Рис. 2. Затримка для тестованих запитів

Витрачений час на графіку 1 тільки з 1 сегментом на запит 1 становить 6 мс. З 5 сегментами, витрачений час становить 4 мс. Аналогічним чином, витрачений час зменшується в міру збільшення кількості сегментів.

Витрачений час на графіку 2 тільки з 1 сегментом на запит 2 становить 12 мс. З 5 сегментами час становить 10 мс. Аналогічним чином, витрачений час зменшується в міру збільшення кількості сегментів.

Витрачений час на графіку 3 тільки з 1 сегментом на запит 3 становить 16 мс. З 5 сегментами час становить 13 мс. Аналогічним чином, витрачений час зменшується в міру збільшення кількості сегментів.

На рис. 3 представлено графіки максимальних оцінок документів за різними наборами сегментів для тестованих запитів.

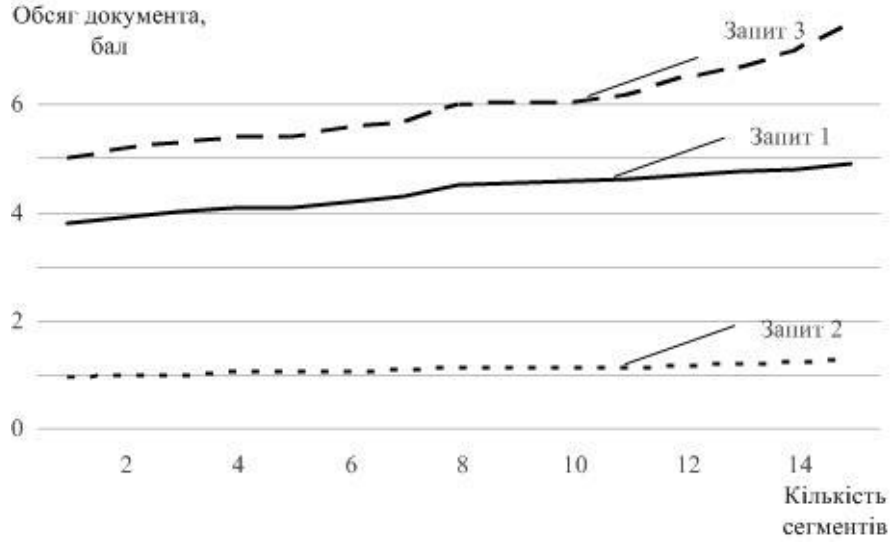

Рис. 3. Максимальні оцінки документів за різними наборами сегментів для тестованих запитів

На рис. 3 для запиту 1 представлено оцінки документів. Максимальний бал з 1 сегментом становить 3,7, а з 5 сегментами - 4,15. Аналогічно, оцінка збільшується в міру збільшення кількості сегментів, кількість витягнутих документів становить 132.

На рис. 3 для запиту 2 представлено оцінки документа. Максимальний бал з 1 сегментом становить 0,95, а з 5 сегментами - 1,07. Оцінка збільшується в міру збільшення кількості сегментів, кількість витягнутих документів становить 96.

На рис. 3 для запиту 3 представлено оцінки документа. Максимальний бал з 1 сегментом становить 5,0, а з 5 сегментами - 5,4. Оцінка збільшується в міру збільшення кількості сегментів, кількість витягнутих документів становить 32.

Таким чином, результати показують, що в міру збільшення кількості сегментів показник подібності запиту документа також збільшується, і для даного запиту витягується найбільш релевантний документ.

В роботі проаналізовано використання методу сегментування з використанням платформи Elasticsearch, який дозволить вирішити питання підвищення масштабованості, узгодженості та відмовостійкості розподілених баз даних та великих пошукових систем. Показано, що такий підхід дасть змогу ранжувати документи, які мають найбільше відношення до запиту користувача. Розглянуто процес вибору сегментів. Показано, що вибір фрагмента відіграє важливу роль у визначенні релевантних результатів для запиту, а також економить ресурси й збільшує час пошуку документів В роботі проведено аналіз затримки часу, витраченого різними наборами сегментів, та проведено оцінку документів за різними наборами сегментів для тестованих запитів.

*Л*ітература

1. Zhuyun D., Chenyan X., Jamie C. Query-Biased Partitioning for Selective Search. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16).* ACM, New York, 2016. P. 1119–1128.

2. Чио К., Фримэн Д. Машинное обучение и безопасность: пер. с англ. А.В. Снастина. М.: ДМК Пресс, 2020. 338 с.

3. Praveen M. Dhulavvagol I., Vijayakumar H. Bhajantri, S. G. Totad Performance Analysis of Distributed Processing System using Shard Selection Techniques on Elasticsearch. *International Conference on Computational Intelligence and Data Science (ICCIDS 2019), Procedia Computer Science*. 2020. Vol. 167. P.1626–1635.