УДК 681.3

К.т.н. Замятін Д. С., студент Михайлюк В. А.

Національний технічний університет України
«Київський політехнічний інститут»

ОЦІНКА ШВИДКОДІЇ МЕТОДІВ ПОВНОТЕКСТОВОГО ПОШУКУ З УРАХУВАННЯМ КЕШУ

Abstract

Denis Zamyatin, PhD; Vadim Mykhailyuk, student

This article examines the inverted index to increase performance by improving the algorithm used to find values in it. The cache-friendly modification of a binary search is proposed. The experimental study and comparison of the performance of a typical binary search and proposed search method is investigated.

Вступ

Широке застосування інформаційних технологій в сучасному суспільстві призвело до створення значних обсягів текстових ресурсів в електронній формі, що ускладнює їх пошук та каталогізацію. Вирішення цієї проблеми можливе за допомогою систем повнотекстового пошуку. Але використання глобальних пошукових систем неможливе в корпоративних сховищах даних в зв'язку з регламентуванням доступу до них, яке необхідне для забезпечення інформаційної захищеності. В той же час, ефективне застосування алгоритмів та методів повнотекстового пошуку для корпоративних ресурсів ускладнюється відсутністю спеціалізованих швидкодіючих обчислювальних засобів.

Отже, оптимізація алгоритмів систем повнотекстового пошуку для локальних корпоративних сховищ з обмеженими обчислювальними ресурсами на сьогодні є актуальною задачею.

У роботі [1] розглянуто підхід до організації інвертованого індексу у пам’яті, які дозволяють підвищити швидкодію. У його основі лежить алгоритм бінарного пошуку, застосування якого пов'язано з проблемою неефективного використання кешу.

Алгоритм бінарного пошуку під час визначення значення центрального елементу кожного наступного пошукового діапазону звертається до сторінок

оперативної пам'яті, які рознесені між собою досить далеко. В результаті, більшість звернень до пам'яті відносяться до комірок, які не можуть бути знайдені у кеші.

Через велику кількість «промахів» кешу швидкодія суттєво нижче від потенційно можливої. У роботах [2, 3] запропоновано варіант прискорення пошуку шляхом організації додаткової «кешуючої» структури. Недолік такого підходу— збільшення часу вставки. Але враховуючи те, що пріоритетною задачею даної системи є пошук, час вставки не є критичним.

Постановка задачі

Отже, задача полягає у вдосконаленні бінарного пошуку з урахуванням можливостей оптимізації кешування.

Архітектура пошукової системи

На основі аналізу даних [2, 3] робіт можна запропонувати наступну архітектуру пошукової системи. Система складається з одного потоку-менеджера, який виконує пошук по допоміжній структурі, і пула потоків, прив’язаних до процесорів. Останні продовжують пошук по основному масиву даних, користуючись результатами попереднього визначення діапазону пошуку, отриманими потоком-менеджером.

Структури даних складаються з основного масиву даних та допоміжної структури, яка має повністю вміщатися у кеші процесора. У цій структурі зберігаються значення елементів основного масиву, які розташовані по індексам, за якими звичайно виконуються порівняння при бінарному пошуку: посередині, через четверту частину масиву и т. д.

Оцінка продуктивності вдосконаленого методу

Для більшості сучасних обчислювальних систем можна вважати вірним припущення, що швидкість читання даних, що присутні у кеші, більша за швидкість читання даних з «промахом» кеша більше, ніж у 3 рази. Проведені розрахунки з взятим мінімальним значенням переваги кешу, у випадку масивів 2-х байтних значень, 4*K* сторінки кеш-пам’яті дали формулу

$$W = \frac{3\log_2 N - 23}{\log_2 N + 1}$$

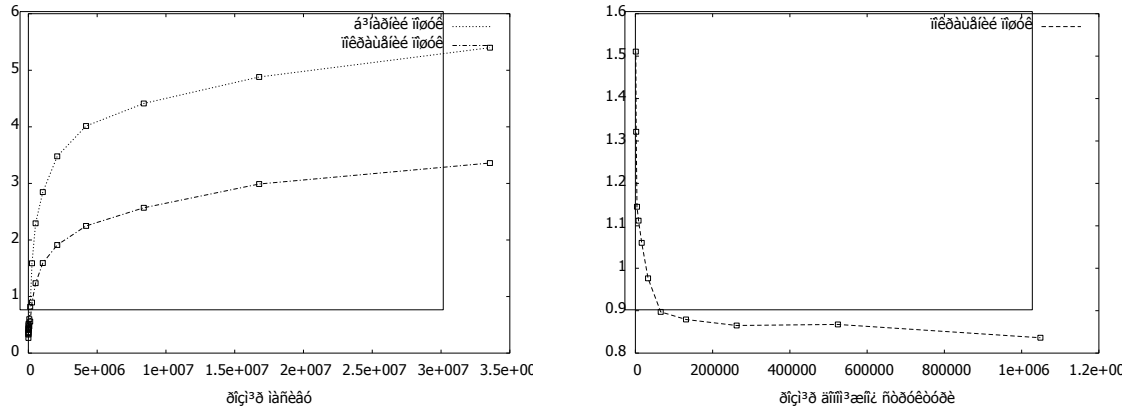
Для додаткового прискорення можна застосовувати векторні інструкції (зокрема, SSE2, Altivec, а також архітектура CUDA), які нададуть можливість виконувати одночасно декілька операцій порівняння елементів з допоміжної структури.

Експериментальне дослідження

Експериментальне дослідження приведених залежностей було виконано на спрощеному однопоточному зразку пошукової системи.

При виконанні пошуку спочатку перебираються значення з допоміжної структури. При цьому на кожній наступній ітерації виконується перехід до пари елементів з вдвічі більшим індексом. Якщо попереднє значення було більше за шукане, з пари вибирається перший елемент, і навпаки. У випадку рівності замість переходу закінчується процедура пошуку. Якщо допоміжна структура вичерпується, пошук продовжується методом звичайного бінарного пошуку на основній.

Результати проведеного вимірювання часу, який займає 1000000 ітерацій пошуку випадково обраного значення у масиві випадкових 2-байтних цілих наведено на графіках. Використане апаратне забезпечення - комп'ютер з процесором *Intel Pentium* 4 1.80*Ghz,* 256 *MB RAM*.



а) б)

Рис. 1. Графіки залежностей часу пошуку від розміру основного масиву та часу пошуку від розміру допоміжної структури.

Аналіз результатів

Розглянемо графік залежності часу виконання пошуку обома методами від розміру масиву (рис. 1а). На початку графіку обсяг масиву невеликий і

повністю поміщається у декілька сторінок кешу. Тому різниця у швидкодії між розглянутими методами невелика і деяка перевага вдосконаленого пошуку пояснюється кращим використанням кешу верхнього рівня за рахунок близького розташування у допоміжній структурі значень, що зчитуються. Далі різниця значно збільшується. Загалом спостерігається залежність, близька до логарифмічної, що дозволяє говорити про відповідність розрахунків експериментально отриманим даним.

Розглянемо графік, на якому показана залежність часу виконання пошуку від розміру допоміжної структури даних (рис. 1б). На ньому видно, що збільшення обсягу допоміжної структури даних дозволяє значно прискорити виконання пошуку. Але починаючи з певної точки подальше нарощення структури дає все менший і менший ріст швидкодії. Тому доцільно обрати розмір, який дає найбільше відношення збільшення швидкодії до обсягу додатково зайнятої пам'яті.

Висновок

Запропоноване вдосконалення пошукового індексу на основі структур даних, що добре кешуються, дозволяє значно підвищити продуктивність пошуку за рахунок ефективного використання кешу. Проведене тестування ефективності на масиві випадкових даних свідчить про збільшення швидкості пошуку у 1,2-1,6 разів.

Література

1. *Замятін Д.С., Михайлюк В.А., Петрашенко А.В.* Підвищення продуктивності повнотекстового пошуку шляхом реорганізації подання інвертованих індексів // Науковий вісник Чернівецького університету. Збірник наукових праць. Випуск 426. — Чернівці. — 2008. — С. 63-67.
2. *Gerth Stølting Brodal, Rolf Fagerberg, Riko Jacob.* Cache Oblivious Search Trees via Binary Trees of Small Height. — BRICS Report Series RS-01-36. — 2001. — 23 p.
3. *Michael A. Bender, Gerth Stølting Brodal, Rolf Fagerberg, Dongdong Ge, Simai He, Haodong Hu, John Iacono, and Alejandro Lopez-Ortiz.* The cost of cache-oblivious searching. In Proceedings of the 44st Annual Symposium on Foundations of Computer Science, Cambridge, Massachusetts. — October 2003. — P. 271-282p.