УДК 004.8

А.А. Суржок
В.О. Денисюк

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ RADIX SORT

Вінницький національний технічний університет

Анотація

У роботі розглянуто розробку паралельного алгоритму сортування за розрядами (Radix Sort) на базі багатопотокового виконання в середовищі Java. Проаналізовано класичні методи сортування, їх обмеження при обробці великих масивів даних, та обґрунтовано вибір Radix Sort як алгоритму з лінійною складністю O(n·k). Описано архітектуру програмного модуля: розподіл обчислень поцифрового сортування між потоками за допомогою ExecutorService та механізмів синхронізації (CountDownLatch). Виконано програмну реалізацію як послідовного, так і паралельного алгоритмів на Java і проведено експериментальне тестування на масивах різного розміру. Результати показують, що паралельний підхід забезпечує коректність сортування та суттєве прискорення при великих обсягах даних, водночас накладні витрати на керування потоками амортизуються із ростом розміру масиву.

Ключові слова: *сортування за розрядами, Radix Sort, Java, багатопоточність, паралельні обчислення, продуктивність.*

Abstract

The paper discusses the development of a parallel algorithm for sorting by digits (Radix Sort) based on multithreaded execution in the Java environment. Classical sorting methods are analyzed, their limitations when processing large data sets, and the choice of Radix Sort as an algorithm with linear complexity O(n·k) is justified. The architecture of the software module is described: distribution of digit sorting calculations between threads using ExecutorService and synchronization mechanisms (CountDownLatch). Software implementation of both sequential and parallel algorithms in Java is performed and experimental testing is conducted on arrays of different sizes. The results show that the parallel approach ensures correct sorting and significant acceleration for large data volumes, while the overhead of thread management is amortized with the growth of the array size.

Keywords*: sort by digits, Radix Sort, Java, multithreading, parallel computing, performance.*

Вступ

Сортування є однією з найпоширеніших операцій у програмних системах баз даних, обробки великих даних, графіці, машинному навчанні та ін. Класичні алгоритми порівняльного сортування (QuickSort, MergeSort, HeapSort) мають часову складність $O(n \log n)$ і при збільшенні обсягу даних стають вузьким місцем [1]. Сортування за розрядами (Radix Sort) відрізняється тим, що не порівнює елементи між собою, а розподіляє їх у «кошику» за окремими розрядами ключів [2]. Радиксний алгоритм виконує серію проходів по цифровому представленню чисел, використовуючи стабільні допоміжні сортування (наприклад, підрахункове сортування, counting sort) на кожному розряді. Часова складність сортування за розрядами $O(wn)$ для n ключів, цілих розміром в машинне слово w. Іноді w представляють як сталу, що може зробити сортування за розрядами привабливішим (для достатньо великого n). Проте, послідовна реалізація Radix Sort не повною мірою використовує можливості багатоядерних процесорів і може бути обмежена затримками при багаторазовому проході по пам’яті. Саме тому актуальним є застосування паралельних технологій: реалізація Radix Sort з одночасним виконанням підзадач на кількох потоках CPU дозволяє суттєво скоротити час сортування великих даних [3, 4].

Постановка задачі дослідження

Основні задачі роботи полягали в розв’язанні наступних питань:

- аналіз існуючих алгоритмів сортування та оцінка їх придатності до паралельної реалізації;
- обґрунтування вибору алгоритму Radix Sort як базового методу для нерівняльного сортування;
- обґрунтування вибору Java як мови програмування і засобів багатопотоковості (наприклад, ***ExecutorService***, синхронізаторів) для прискорення обчислень;
- проектування архітектури паралельної реалізації Radix Sort, включаючи розподіл функцій між потоками;
- розробка і реалізація послідовного та багатопотокового алгоритмів Radix Sort мовою Java;
- проведення експериментального тестування продуктивності реалізації на масивах різної довжини та аналіз отриманих результатів.

Виклад основного матеріалу

Алгоритми сортування є одними з базових процедур комп'ютерної науки та широко застосовуються в інформаційних системах, базах даних, системах аналізу великих даних, комп'ютерній графіці та мережевих застосунках [1–4]. Серед класичних методів сортування розрізняють алгоритми, засновані на порівняннях (QuickSort, MergeSort, HeapSort), для яких характерна часова складність порядку $O(n\log n)$, а також лінійні алгоритми, зокрема сортування за розрядами (Radix Sort), складність якого визначається як *O(wn)* і є ефективною для впорядкування числових масивів із фіксованою розрядністю ключів. На основі аналізу літературних джерел було досліджено принципи роботи алгоритму Radix Sort, зокрема поетапне поцифрове опрацювання елементів, стабільність допоміжного сортування та залежність продуктивності від кількості розрядів і розміру вхідного масиву. Окрему увагу приділено питанням паралельної обробки даних на багатоядерних процесорах, а також можливостям використання засобів багатопоточності мови Java для підвищення продуктивності алгоритмів сортування [5]. У роботі розглянуто паралельну реалізацію алгоритму сортування за розрядами, побудовану на використанні центрального процесора та механізмів багатопотокового виконання. Основна ідея методу полягає у розподілі обчислень між кількома потоками виконання, що дозволяє одночасно обробляти різні частини вхідного масиву. Паралелізація застосовується на етапах побудови гістограм значень поточного розряду та стабільного перерозподілу елементів, що є найбільш обчислювально затратними фазами алгоритму Radix Sort.

Програмна реалізація виконана мовою Java з використанням стандартних засобів пакета java.util.concurrent, зокрема пулу потоків ExecutorService та механізмів синхронізації. Вхідний масив ділиться на рівні сегменти відповідно до кількості потоків, після чого кожен потік незалежно обробляє свій сегмент, формуючи локальну гістограму частот значень поточного розряду. Такий підхід дозволяє уникнути конфліктів доступу до спільних даних та зменшити накладні витрати на синхронізацію. Після завершення побудови локальних гістограм виконується їх об'єднання у глобальну гістограму, на основі якої обчислюється кумулятивна сума, що визначає позиції елементів у відсортованому масиві. Подальший етап стабільного перерозподілу також виконується паралельно: кожен потік переносить елементи зі свого сегмента у допоміжний масив відповідно до попередньо обчислених зсувів. Синхронізація між етапами забезпечує коректність обчислень та узгодженість доступу до пам'яті. Для оцінки ефективності паралельної реалізації алгоритму було проведено серію експериментів із масивами випадкових чисел різного розміру. У процесі вимірювався загальний час виконання алгоритму Radix Sort, що включає всі проходи по розрядах, а також перевірялась коректність сортування за допомогою стандартних засобів перевірки впорядкованості масиву. Отримані результати свідчать, що при збільшенні розміру вхідних даних спостерігається відносне зменшення часу обробки одного елемента, що пояснюється більш повним завантаженням обчислювальних ресурсів процесора та амортизацією накладних витрат на керування потоками (табл.1, 2).

Таким чином, застосування паралельної реалізації Radix Sort на мові Java є доцільним для сортування великих числових масивів на багатоядерних процесорах. Запропонований підхід дозволяє

підвищити продуктивність алгоритму без використання спеціалізованих графічних обчислень, спираючись виключно на можливості стандартних засобів багатопоточності центрального процесора.

Таблиця 1 – Результати тестування реалізацій Radix Sort

Кількість елементів	Послідовна реалізація у одному потоці, мс	Паралельна реалізація у потоках, мс			
		2 потоки	потоки	потоків	потоків

Висновки

У ході виконання роботи було розглянуто та проаналізовано алгоритм сортування за розрядами (Radix Sort) як один із ефективних лінійних методів упорядкування числових даних. На основі аналізу літературних джерел встановлено, що Radix Sort не використовує операції порівняння між елементами, а базується на поетапному поцифровому опрацюванні ключів, що забезпечує часову складність порядку $O(wn)$ за умови фіксованої кількості розрядів.

У роботі запропоновано та реалізовано паралельний підхід до виконання алгоритму Radix Sort із використанням засобів багатопоточності мови Java. Основна ідея методу полягає у розподілі вхідного масиву на сегменти та паралельному виконанні найбільш обчислювально затратних етапів алгоритму, зокрема побудови гістограм значень розрядів і стабільного перерозподілу елементів. Застосування пулу потоків та механізмів синхронізації дозволяє забезпечити коректність обчислень і узгоджений доступ до спільних ресурсів пам’яті.

Експериментальні дослідження підтвердили, що при зростанні розміру вхідних масивів паралельна реалізація демонструє кращу продуктивність порівняно з послідовним варіантом алгоритму. Це пояснюється більш повним використанням обчислювальних ресурсів багатоядерного центрального процесора та зменшенням впливу накладних витрат на керування потоками при обробці великих обсягів даних.

Отримані результати свідчать про доцільність застосування паралельної реалізації Radix Sort на мові Java для задач сортування великих числових масивів у програмних системах загального призначення. Запропонований підхід може бути використаний як основа для подальших досліджень у напрямі оптимізації алгоритмів сортування та розробки високопродуктивних програмних рішень на багатоядерних платформах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кнут Д. Мистецтво програмування. Том 3. Сортування та пошук. М.: Вільямс, 2017.
2. Radix Sort. URL: https://www.geeksforgeeks.org/dsa/radix-sort/
3. Grama A., Gupta A., Karypis G., Kumar V. Introduction to Parallel Computing. Addison-Wesley, 2003.
4. McCool M., Robison A., Reinders J. Structured Parallel Programming. Morgan Kaufmann, 2012.
5. Oracle. Java Concurrency and Parallelism Documentation.

Суржок Антон Андрійович – студент кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: anton.surzhok@gmail.com

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп’ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Surzhok Anton Andriyovych – student of the Department of Computer Science, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: anton.surzhok@gmail.com

Denysiuk Valeriy Oleksandrovych – Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua.