

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ STRUCTURE SORTING

Вінницький національний технічний університет

Анотація

Розглянуто розробку та програмну реалізацію оптимізованого паралельного алгоритму сортування класу Structure Sorting на основі Parallel QuickSort із використанням технологій платформи .NET та мови програмування C#. Проаналізовано роль паралельного сортування в сучасних комп'ютерних системах, виконано математичне моделювання та структурний аналіз алгоритму, синтезовано його потоковий граф. Обґрунтовано вибір програмних засобів реалізації, розроблено UML-діаграму класів і блок-схему алгоритму. Створено програмну реалізацію оптимізованого паралельного алгоритму та проведено тестування його продуктивності для масивів різного розміру. Використання отриманих результатів дозволяє підвищити швидкодію, продуктивність та ефективність сортування великих обсягів даних у багатопотокових обчислювальних середовищах.

Ключові слова: паралельний алгоритм, паралельне сортування, Structure Sorting, Parallel QuickSort, потоковий граф, C#.

Abstract

The development and software implementation of an optimized parallel sorting algorithm of the Structure Sorting class based on Parallel QuickSort using the .NET platform and the C# programming language are considered. The role of parallel sorting in modern computer systems is analyzed, mathematical modeling and structural analysis of the algorithm are performed, and its flow graph is synthesized. The choice of software implementation tools is justified, and a UML class diagram and a block diagram of the algorithm are developed. A software implementation of the optimized parallel algorithm is created, and its performance is tested on data arrays of various sizes. The obtained results can be used to improve the speed, productivity, and efficiency of sorting large data sets in multi-threaded computing environments.

Keywords: parallel algorithm, parallel sorting, Structure Sorting, Parallel QuickSort, flow graph, C#.

ВСТУП

Актуальність паралельного сортування структур даних зумовлена стрімким зростанням обсягів інформації та широким використанням багатоядерних обчислювальних систем. У сучасних програмних комплексах, зокрема в системах Big Data, базах даних, фінансових застосунках та обчислювальних платформах, послідовні алгоритми сортування часто не забезпечують необхідного рівня продуктивності. Використання паралельних алгоритмів дозволяє ефективніше задіяти обчислювальні ресурси та скоротити час виконання операцій над великими масивами даних.

Метою роботи є розробка, структурний аналіз та програмна реалізація оптимізованого паралельного алгоритму сортування класу Structure Sorting на основі Parallel QuickSort із використанням технологій .NET та мови програмування C# [1-4].

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Для досягнення поставленої мети у роботі необхідно виконати аналіз предметної області паралельного сортування, здійснити математичне моделювання алгоритму Parallel QuickSort, синтезувати його потоковий граф, спроектувати програмну реалізацію з використанням UML-діаграм, реалізувати оптимізований паралельний алгоритм та провести тестування з аналізом отриманих результатів.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Паралельні алгоритми сортування є важливою складовою сучасних обчислювальних систем, оскільки дозволяють значно зменшити час обробки великих масивів даних за рахунок розпаралелювання обчислень. У роботі розглянуто алгоритм Parallel QuickSort, який базується на класичному алгоритмі швидкого сортування та використовує модель fork/join для створення паралельних підзадач [1, 2].

На основі математичного моделювання досліджено часову складність алгоритму, сформульовано рекурентні співвідношення часу виконання, визначено критичний шлях, висоту та ширину потокового графу, а також потенційний ступінь паралельності. Показано, що за умови збалансованого вибору опорного елемента алгоритм зберігає асимптотичну складність порядку $O(n \log n)$ і характеризується високим потенціалом ефективного розпаралелювання.

У рамках структурного аналізу виконано синтез потокового графу алгоритму Parallel QuickSort із використанням матриці суміжності та інших матричних подань, що дозволило формально оцінити можливості оптимізації та зменшення накладних витрат [3]. Для програмної реалізації розроблено UML-діаграму класів та блок-схему алгоритму, які відображають логіку рекурсивної обробки даних, вибору опорного елемента, переходу між паралельним і послідовним режимами сортування та синхронізації результатів.

Алгоритм реалізовано мовою C# з використанням засобів Task Parallel Library. З метою підвищення ефективності застосовано пороговий розмір підмасиву для переходу до послідовного сортування, обмеження глибини паралельної рекурсії, а також винесення стратегії вибору опорного елемента в окремий компонент [3, 4].

Висновки

Проведене тестування програмної реалізації підтвердило коректність роботи оптимізованого паралельного алгоритму сортування та його придатність до використання в багатопотокових середовищах. Встановлено, що при обробці великих масивів даних паралельна реалізація забезпечує скорочення часу виконання у порівнянні з послідовним варіантом, тоді як застосування порогових умов дозволяє уникнути надлишкових накладних витрат для малих обсягів даних.

Отже, у роботі розв'язано всі поставлені завдання: виконано аналіз предметної області паралельного сортування, здійснено математичне моделювання та структурний аналіз алгоритму класу Structure Sorting на основі Parallel QuickSort, синтезовано його потоковий граф, побудовано UML-діаграми та блок-схеми, а також розроблено й досліджено програмну реалізацію оптимізованого паралельного алгоритму сортування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sorting array of structs on multiple elements in c#. URL: <https://stackoverflow.com/questions/20103261/sorting-array-of-structs-on-multiple-elements-in-c-sharp>
2. How to implement Parallel Quick Sort in C#. URL: <https://www.codeguru.co.in/2023/05/how-to-implement-parallel-quick-sort-in.html>
3. Denysiuk V.O. Review of the Parallel Hyperquick Sort Algorithm by C#. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/41377/144807.pdf?sequence=2&isAllowed=y>
4. Sorting Algorithms in .NET. URL: <https://visualstudiomagazine.com/articles/2004/11/01/sort-in-net.aspx>

Копицяк Станіслав Вікторович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: StasKopicak@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.

Kushnir Oleksandr Anatoliyovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: StasKopicak@gmail.com;

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua