УДК 004.8

М.Я. Бондар
В.О. Денисюк

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ ПОШУКУ DEPTH-FIRST SEARCH

Вінницький національний технічний університет

Анотація

Розроблено оптимізований паралельний алгоритм пошуку в глибину Depth-First Search для графів. Проведено детальний аналіз існуючих методів пошуку в глибину, обґрунтовано вибір засобів для реалізації паралельного виконання та моделювання часових затримок між вузлами графу. Визначено переваги та недоліки класичних підходів та запропоновано методи їхньої оптимізації. Запропоновано використання бібліотеки ThreadPoolExecutor у Python для реалізації багатопотокового обчислення. Особлива увага приділена ефективності використання потоків, балансуванню навантаження та мінімізації синхронізаційних витрат. Створено програмну реалізацію паралельного Depth-First Search, що дозволяє значно скоротити час виконання алгоритму, особливо на великих графах. Проведене тестування підтвердило зменшення часу виконання на 1.5-2 рази в порівнянні з послідовною версією, що підтверджує ефективність багатопотокового підходу. Отримані результати можуть бути використані в задачах реального часу, аналізі соціальних мереж, оптимізації маршрутів, а також у розподілених системах обчислень.

Ключові слова: *пошук в глибину, паралельний алгоритм, багатопоточність, граф.*

Abstract

An optimized parallel Depth-First Search algorithm for graphs has been developed. A detailed analysis of existing DFS methods was conducted, and the choice of tools for parallel execution and modeling of time delays between graph nodes was justified. The advantages and disadvantages of classical approaches were identified, and methods for their optimization were proposed. The use of the ThreadPoolExecutor library in Python was suggested for implementing multithreaded computation. Special attention was given to thread efficiency, load balancing, and minimizing synchronization overhead. A software implementation of the parallel Depth-First Search algorithm was created, significantly reducing execution time, especially for large graphs. Testing confirmed a reduction in execution time by 1.5-2 times compared to the sequential version, demonstrating the effectiveness of the multithreading approach. The obtained results can be applied to real-time tasks, social network analysis, route optimization, and distributed computing systems.

Keywords: *depth-first search, parallel algorithm, multithreading, graph.*

Вступ

Розвиток комп'ютерних систем із багатоядерними процесорами створює можливості для використання паралельних алгоритмів [1]. Алгоритм пошуку в глибину Depth-First Search (DFS) широко застосовується в задачах обходу графів, пошуку шляхів та перевірки зв'язності. Проте його традиційна послідовна реалізація може бути неефективною для великих графових структур [2]. У роботі розглянуто підходи до оптимізації DFS шляхом багатопотокового виконання для підвищення продуктивності алгоритму.

Постановка задачі дослідження

Для реалізації паралельного алгоритму пошуку в глибину необхідно [3]:

- проаналізувати існуючі методи реалізації DFS [4];
- розробити та оптимізувати паралельну версію DFS;
- виконати тестування продуктивності та порівняти результати з послідовною реалізацією.

Виклад основного матеріалу

Розглянуто класичний алгоритм DFS та його паралельні модифікації. Запропоновано підхід до розподілу обчислень між потоками з використанням ThreadPoolExecutor у Python [5-6]. Проведено оптимізацію, що включає зменшення накладних витрат на синхронізацію, використання структур даних для ефективного управління відвіданими вузлами та обмеження кількості активних потоків. Реалізація дозволила суттєво зменшити час виконання алгоритму для великих графів.

Результати дослідження.

Проведене тестування паралельного DFS показало зменшення часу виконання в середньому на 1.5-2 рази порівняно з послідовною версією [7]. Найбільший ефект спостерігався при обробці графів із великою кількістю вершин. Аналіз впливу кількості потоків на продуктивність підтвердив, що оптимальне їхнє число залежить від апаратних характеристик системи.

Висновки

Результати дослідження підтвердили ефективність застосування паралельної обробки для DFS. Використання багатопоточності дозволяє значно скоротити час обчислень, що особливо важливо для аналізу великих графових структур. Подальші дослідження можуть бути зосереджені на використанні GPU для ще більшого прискорення обчислень та розподіленої обробки графів у кластерних системах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сухий О. Л. Алгоритми пошуку в інформаційних системах: методичні рекомендації / О.Л.Сухий, В.М.Міленін, В.М.Тарадайнік. К., 2015. 71с.
2. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн, Вступ до алгоритмів. К. : К. І. С., 2019. 1288 с.
3. Sedgewick, R., & Wayne, K. Algorithms. 4th Edition. Addison-Wesley, 2011. 955с.
4. Креневич А.П. Алгоритми і структури даних. Підручник. К.: ВПЦ "Київський Університет", 2021. 200 с.
5. А. В. Яковенко. Основи програмування. Python. Частина 1: підручник для студ. спеціальності 122 «Комп'ютерні науки», спеціалізації «Інформаційні технології в біології та медицині»; Київ : КПІ ім. Ігоря Сікорського, 2018. 195 с.
6. Довідник з мови Python. [URL]: https://docs.python.org/uk/3/reference/index.html
7. Depth First Search. [URL]: https://www.hackerearth.com/practice/algorithms/graphs/depth-first search/tutorial/

Бондар Микола Ярославович – студент кафедри комп`ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: koliabondar2008@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.

Bondar Mykola Yaroslavovich – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: koliabondar2008@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