

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ СОРТУВАННЯ BOGOSORT ІЗ ВИКОРИСТАННЯМ OPENMP

Вінницький національний технічний університет

Анотація

Розглянуто розробку та програмну реалізацію паралельного алгоритму сортування BogoSort з використанням технології OpenMP. Виконано аналіз принципу роботи BogoSort як стохастичного алгоритму сортування, обґрунтовано доцільність застосування паралельних обчислень для прискорення процесу пошуку впорядкованої перестановки. Описано підхід до реалізації багатопотокової версії алгоритму, де кожен потік незалежно генерує випадкові перестановки та перевіряє їх на впорядкованість із використанням спільного прапорця завершення. Проведено тестування продуктивності послідовної та паралельної реалізацій для різних розмірів масивів і різної кількості потоків. Отримані результати демонструють зменшення часу виконання при використанні паралелізму, однак показують обмежену масштабованість через накладні витрати синхронізації та випадковий характер алгоритму. Практичне значення роботи полягає у демонстрації впливу багатопотоковості на швидкодію та вивченні особливостей паралельних стохастичних алгоритмів.

Ключові слова: BogoSort, сортування, OpenMP, паралельні обчислення, багатопотоковість, продуктивність.

Abstract

This paper considers the development and software implementation of the parallel BogoSort algorithm using OpenMP. The principle of BogoSort as a stochastic sorting method is analyzed, and the use of parallel computing is justified to accelerate the search for a sorted permutation. A multithreaded approach is presented where each thread independently generates random permutations and checks sortedness using a shared termination flag. Performance testing of sequential and parallel versions is carried out for different array sizes and thread counts. The results show reduced execution time with parallelism, while also highlighting limited scalability due to synchronization overhead and the randomized nature of the algorithm. The work is useful for demonstrating the impact of multithreading on performance and for studying parallel stochastic algorithms.

Keywords: BogoSort, sorting, OpenMP, parallel computing, multithreading, performance.

Вступ

Актуальність дослідження паралельної реалізації BogoSort полягає у можливості наочно продемонструвати вплив багатопотоковості на час виконання алгоритмів із випадковою природою. BogoSort є відомим прикладом неефективного алгоритму сортування, який базується на багаторазовому випадковому перемішуванні масиву до моменту отримання впорядкованого стану [1, 2]. Хоча на практиці цей алгоритм не застосовується через факторіальну складність, він є зручним для навчальних цілей, аналізу й експериментів із паралельними обчисленнями. Використання OpenMP дозволяє організувати паралельні спроби пошуку відсортованої перестановки, що потенційно зменшує очікуваний час роботи за рахунок одночасного виконання незалежних ітерацій у кількох потоках.

Метою роботи є розробка послідовної та паралельної реалізацій BogoSort, аналіз їх продуктивності та оцінювання ефекту від використання різної кількості потоків.

Постановка задачі дослідження

Задачі дослідження полягають у реалізації послідовної версії BogoSort, розробці паралельного варіанту з використанням OpenMP, організації коректного завершення всіх потоків після знаходження відсортованого масиву, проведенні тестування для різних розмірів вхідних даних та

різної кількості потоків, а також у розрахунку показників прискорення та ефективності паралельної реалізації [1-3].

Виклад основного матеріалу

BogoSort (Permutation Sort) працює за принципом «перемішати та перевірити». На кожній ітерації виконується перевірка, чи є масив відсортованим за зростанням. Якщо масив невпорядкований, генерується нова випадкова перестановка елементів, після чого перевірка повторюється. У загальному випадку очікувана кількість спроб до появи відсортованої перестановки зростає факторіально із збільшенням розміру масиву, що робить алгоритм непридатним для практичного використання.

Паралельна реалізація базується на ідеї незалежних спроб: кожен потік отримує власну копію масиву, виконує цикл випадкових перестановок і перевірок, а при успіху встановлює спільний прапорець `globalFound = true`. Після встановлення прапорця інші потоки припиняють роботу під час наступної перевірки умови завершення. Такий підхід мінімізує взаємозалежність потоків, адже основні обчислення виконуються незалежно, а синхронізація зводиться до перевірки та встановлення спільного логічного значення.

Для оцінювання продуктивності використано показники прискорення та ефективності. Коефіцієнт прискорення визначається як $S = T_{seq}/T_{par}$, де T_{seq} — час послідовної реалізації, T_{par} — час паралельної реалізації. Коефіцієнт ефективності обчислюється як $E = S/p$, де p — кількість потоків (табл.1-3).

Таблиця 1 – Час виконання програми на різних вхідних даних

Кількість потоків	n = 5 000 (с)	n = 10 000 (с)
Послідовно	447.607	905.932
2	251.682	496.471
4	309.426	565.272

Таблиця 2 – Коефіцієнти прискорення S

Кількість потоків	n = 5 000	n = 10 000
2	1.7785	1.8247
4	1.5017	1.8247

Таблиця 3 – Коефіцієнти ефективності E

Кількість потоків	n = 5 000	n = 10 000
2	0.8892	0.9124
4	0.3754	0.4051

Проаналізувавши час виконання (табл. 1) та розраховані коефіцієнти прискорення і ефективності (табл. 2–3), можна зробити висновок, що паралельна реалізація забезпечує помітне зменшення часу виконання порівняно з послідовною версією. Для $n = 5000$ використання 2 потоків дає прискорення близько 1.78, а для $n = 10000$ — близько 1.82. При збільшенні кількості потоків до 4 прискорення зростає не пропорційно, що пояснюється накладними витратами синхронізації та обмеженнями апаратної платформи.

Важливо враховувати випадковий характер BogoSort: тривалість роботи може суттєво залежати від кількості ітерацій до знаходження відсортованої перестановки, тому значення часу мають

природні коливання. Водночас у проведених вимірюваннях зберігається загальна тенденція: паралельний підхід зменшує очікуваний час виконання, але не усуває факторіальну складність алгоритму. Збільшення числа потоків понад певне значення може давати менший вигравш через витрати на керування потоками та конкуренцію за ресурси процесора.

Висновки

Розроблено послідовну та паралельну реалізації алгоритму BogoSort із використанням OpenMP. Проведено тестування для $n = 5000$ та $n = 10000$ при 2 і 4 потоках та розраховано показники прискорення й ефективності. Встановлено, що паралельна реалізація дає помітне зменшення часу виконання, однак масштабованість обмежується накладними витратами та випадковою природою алгоритму. Загалом BogoSort є доцільним як навчальний приклад для демонстрації принципів паралельних обчислень і оцінювання продуктивності багатопотокових програм.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. BogoSort or Permutation Sort. URL: <https://www.geeksforgeeks.org/dsa/bogosort-permutation-sort/>
2. Neto A. Bogosort: The Stupid Sorting Algorithm. URL: <https://dev.to/adolfont/bogosort-the-stupid-sorting-algorithm-168f>
3. OpenMP Architecture Review Board. OpenMP Books and Resources URL: <https://www.openmp.org/resources/openmp-books/>

Іосіф Владислав Олександрович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: vladgguyt@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.

Iosif Vladyslav Oleksandrovych – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: vladgguyt@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