

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ БУЛЬБАШКОВОГО СОРТУВАННЯ ЗА ДОПОМОГОЮ PYTHON

Вінницький національний технічний університет

Анотація

Розглянуто розробку паралельного алгоритму сортування методом «парне-непарне» (Odd-Even Transposition Sort) з використанням мови програмування Python та бібліотеки multiprocessing. Проведено аналіз ефективності класичного бульбашкового сортування та його паралельної модифікації. Обґрунтовано вибір засобів реалізації, описано архітектуру програми з використанням спільної пам'яті та бар'єрної синхронізації. У роботі створено програмну реалізацію та проведено тестування продуктивності на різних наборах даних, визначено коефіцієнти прискорення та ефективності.

Ключові слова: паралельні обчислення, сортування, Python, multiprocessing, Odd-Even Sort, продуктивність.

Abstract

The development of a parallel sorting algorithm using the Odd-Even Transposition Sort method using the Python programming language and the multiprocessing library is considered. The efficiency of classical bubble sort and its parallel modification is analyzed. The choice of implementation tools is justified, the program architecture using shared memory and barrier synchronization is described. In the work, a software implementation was created and performance testing was carried out on various data sets, acceleration and efficiency coefficients were determined.

Keywords: parallel computing, sorting, Python, multiprocessing, Odd-Even Sort, performance.

Вступ

Актуальність теми полягає у необхідності підвищення швидкодії обробки великих масивів даних у сучасних розподілених системах. Сортування є однією з найбільш ресурсомістких операцій у базах даних та системах аналізу Big Data. Хоча класичне бульбашкове сортування має квадратичну складність $O(N^2)$ і рідко застосовується для великих даних у послідовному вигляді, його модифікація — парне-непарне сортування (Odd-Even Sort) — ідеально підходить для паралельних архітектур завдяки локальності операцій обміну.

Використання можливостей сучасних багатоядерних процесорів дозволяє значно скоротити час виконання сортування, розподіливши навантаження між ядрами. Метою роботи є розробка та дослідження ефективності паралельного алгоритму бульбашкового сортування засобами мови Python.

Постановка задачі дослідження

Задачі дослідження полягають у вирішенні наступних питань: аналіз методів розпаралелювання алгоритмів сортування; синтез потокового графу алгоритму Odd-Even Sort; програмна реалізація алгоритму з мінімізацією накладних витрат на міжпроцесну взаємодію; експериментальне дослідження залежності часу виконання та прискорення від розміру вхідних даних.

Виклад основного матеріалу

Алгоритм бульбашкового сортування базується на порівнянні сусідніх елементів. У паралельній версії (Odd-Even Transposition Sort) процес сортування розбивається на дві незалежні

фази, що чергуються.

1. Парна фаза: Порівняння та обмін пар a_i, a_{i+1} для всіх парних i .
2. Непарна фаза: Порівняння та обмін пар $(a_i, a_i + 1)$ для всіх непарних i .

Така декомпозиція дозволяє виконувати $N/2$ операцій порівняння одночасно. Теоретична часова складність паралельного алгоритму становить $O(N)$ при використанні $N/2$ процесорів [1, 2].

Для реалізації обрано мову Python та бібліотеку multiprocessing. На відміну від потоків (threading), процеси в Python мають власні простори пам'яті та обходять обмеження Global Interpreter Lock (GIL), що дозволяє задіяти всі фізичні ядра CPU [3].

Для оптимізації програмної реалізації було використано наступні підходи.

- Спільна пам'ять (multiprocessing.Array): Дозволила уникнути дорогою вартісної серіалізації даних при передачі між процесами.
- Постійні воркери (Persistent Workers): Пул процесів створюється один раз, а синхронізація фаз відбувається через бар'єри (Barrier), що зменшує накладні витрати на створення процесів.
- Ранній вихід (Early Exit): Використання спільного прапорця для зупинки алгоритму, якщо масив вже відсортований.

Проведено тестування розробленої програми на ПК з процесором Intel Core i5 (6 ядер).

Результати вимірювань часу виконання (у секундах) для масивів різної розмірності наведено в таблиці 1.

Таблиця 1 – Часові характеристики алгоритмів

Розмір масиву (N)	Час послідовний (T1), с	Час паралельний (Tp), с	Прискорення (S)
1 000	0.145	0.170	0.85
2 500	0.890	0.420	2.12
5 000	3.650	1.150	3.17
10 000	14.800	3.950	3.75

Аналіз результатів показує, що на малих масивах ($N < 1000$) паралельний алгоритм поступається послідовному через накладні витрати на синхронізацію. Однак, починаючи з $N \approx 1500$, спостерігається стійке прискорення. Для $N = 1000$ коефіцієнт прискорення досягає 3.75, що свідчить про ефективність розпаралелювання.

Висновки

У роботі досліджено та реалізовано паралельний алгоритм сортування Odd-Even Transposition Sort. Встановлено, що використання бібліотеки multiprocessing у поєднанні зі спільною пам'яттю дозволяє ефективно реалізувати паралельні обчислення на Python.

Експериментально підтверджено, що для великих масивів даних паралельний алгоритм забезпечує суттєве зменшення часу виконання порівняно з послідовним методом. Основними факторами, що впливають на ефективність, є розмір вхідних даних та накладні витрати на синхронізацію процесів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Про алгоритми сортування. URL: [https:// foxminded . ua / alhorytmy - sortuvannia /](https://foxminded.ua/alhorytmy-sortuvannia/)
2. Коцовський В.М. Теорія паралельних обчислень: навчальний посібник. Ужгород: ПП«АУТДОР-Шарк», 2021. 188 с.
3. Python 3.12 Documentation: multiprocessing — Process-based parallelism. URL: <https://docs.python.org/3/library/multiprocessing.html>.

Рижов Дмитро Романович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця. e-mail: valerad64@meta.ua ;

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

Ryzhov Dmytro Romanovich – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia. e-mail: valerad64@meta.ua ;

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