

# РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО АЛГОРИТМУ ШВИДКОГО СОРТУВАННЯ

Вінницький національний технічний університет

## **Анотація**

*У роботі розглянуто реалізацію паралельного алгоритму швидкого сортування з використанням багатопоточності мовою програмування Python. Проаналізовано сучасні підходи до організації паралельних обчислень, виконано математичне моделювання паралельного алгоритму з урахуванням законів Амдала та Густавсона, здійснено синтез потокового графу алгоритму. Обґрунтовано вибір програмного середовища та мови програмування для реалізації алгоритму. Розроблено програмну реалізацію паралельного алгоритму швидкого сортування із використанням потоків, побудовано UML-діаграму діяльності алгоритму. Проведено тестування програмної реалізації та аналіз отриманих результатів. Використання розробленого алгоритму дозволяє підвищити продуктивність обробки даних та ефективніше використовувати ресурси багатоядерних обчислювальних систем.*

**Ключові слова:** паралельні обчислення, швидке сортування, паралельний алгоритм, багатопоточність, Python.

## **Abstract**

*The paper considers the implementation of a parallel quick sort algorithm using multithreading in the Python programming language. Modern approaches to the organization of parallel computing are analyzed, mathematical modeling of the parallel algorithm is performed taking into account Amdahl's and Gustafson's laws, and a task flow graph of the algorithm is synthesized. The choice of the programming language and development environment is justified. A software implementation of the parallel quick sort algorithm using threads is developed, and a UML activity diagram of the algorithm is constructed. Program testing and analysis of the obtained results are carried out. The use of the developed algorithm makes it possible to increase data processing performance and efficiently utilize the resources of multicore computing systems.*

**Keywords:** parallel computing, quick sort, parallel algorithm, multithreading, Python.

## **Вступ**

Актуальність дослідження паралельних алгоритмів сортування зумовлена стрімким зростанням обсягів даних та підвищеними вимогами до швидкодії програмних систем. Алгоритми сортування є базовими компонентами багатьох інформаційних систем, зокрема баз даних, систем обробки великих даних, наукових і фінансових застосунків. Алгоритм швидкого сортування є одним з найефективніших алгоритмів сортування завдяки середній складності  $O(n \log n)$ , однак його класична реалізація не дозволяє повною мірою використовувати можливості сучасних багатоядерних процесорів [1, 2].

Використання паралельних обчислень дає змогу суттєво скоротити час виконання алгоритмів за рахунок одночасної обробки незалежних підзадач. Паралельна реалізація алгоритму швидкого сортування базується на принципі «розділяй та володарюй» і дозволяє виконувати сортування підмасивів у декількох потоках [2]. Метою роботи є розробка та дослідження паралельного алгоритму швидкого сортування з використанням багатопоточності для підвищення продуктивності обчислювальних процесів.

## **Постановка задачі дослідження**

Для досягнення поставленої мети у роботі необхідно вирішити такі задачі:

- проаналізувати підходи до організації паралельних обчислень;
- виконати математичне моделювання паралельного алгоритму швидкого сортування;
- здійснити синтез потокового графу паралельного алгоритму;
- розробити програмну реалізацію паралельного алгоритму швидкого сортування мовою Python;

- провести тестування програми та проаналізувати результати її роботи.

### **Виклад основного матеріалу**

Паралельні алгоритми призначені для ефективного використання багатоядерних обчислювальних систем шляхом розподілу задачі на незалежні підзадачі. У роботі розглянуто основні підходи до паралельних обчислень, зокрема паралелізм на рівні задач, даних та інструкцій. Для оцінювання ефективності паралельних алгоритмів використано закон Амдала, який визначає теоретичні обмеження прискорення, та закон Густавсона, що враховує масштабування задачі зі зростанням кількості процесорів.

Алгоритм швидкого сортування ґрунтується на рекурсивному розбитті масиву відносно опорного елемента. У паралельній реалізації після розбиття масиву сортування лівої та правої частин виконується одночасно у різних потоках. Такий підхід дозволяє зменшити загальний час виконання алгоритму для великих обсягів даних. Для оптимізації алгоритму застосовано обмеження кількості потоків, використання порогового значення для переходу до послідовного сортування та зменшення витрат на синхронізацію.

Програмна реалізація паралельного алгоритму швидкого сортування виконана мовою Python із використанням бібліотеки `concurrent.futures` [3]. Побудовано UML-діаграму діяльності, яка наочно відображає логіку роботи алгоритму. Проведено тестування програми для різних розмірів масивів, що підтвердило коректність роботи алгоритму та доцільність використання паралельного підходу для підвищення продуктивності.

### **Висновки**

У роботі досліджено та реалізовано паралельний алгоритм швидкого сортування з використанням багатопоточності. Проаналізовано теоретичні основи паралельних обчислень, виконано математичне моделювання алгоритму та синтез потокового графу. Розроблено програмну реалізацію паралельного алгоритму мовою Python та проведено її тестування. Отримані результати підтверджують, що паралельний підхід дозволяє ефективніше використовувати ресурси багатоядерних процесорів і зменшити час сортування великих масивів даних. Результати роботи можуть бути використані у навчальних та практичних задачах, пов'язаних із паралельною обробкою даних.

### **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. How to implement Parallel Quick Sort in C#. URL: <https://www.codeguru.co.in/2023/05/how-to-implement-parallel-quick-sort-in.html>
2. 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>
3. Python Threading. URL: <https://docs.python.org/3/library/threading.html>

**Гончар Володимир Юрійович** – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця. e-mail: vovan4uk.2005@gmail.com ;

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

**Honchar Volodymyr Yuriyovych** – student of the Department of Computer Science, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia. e-mail: vovan4uk.2005@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