

ДОСЛІДЖЕННЯ СОРТУВАННЯ ВЕКТОРА В РОЗПОДІЛЕНІЙ СИСТЕМІ

Вінницький національний технічний університет

Анотація

У роботі розглянуто реалізацію паралельного алгоритму сортування вектора в розподіленій системі. Проаналізовано архітектуру Master–Worker, розроблено програмну реалізацію алгоритму та проведено тестування його продуктивності для різних розмірів вхідних даних і кількості процесів. Досліджено залежність часу виконання, коефіцієнтів прискорення та ефективності від параметрів системи, що дозволило оцінити доцільність використання паралельного сортування для обробки великих обсягів даних.

Ключові слова: розподілена система, паралельний алгоритм, сортування вектора, Master–Worker, продуктивність.

Abstract

The paper considers the implementation of a parallel vector sorting algorithm in a distributed system. The Master–Worker architecture is analyzed, a software implementation is developed, and performance testing is conducted for different input sizes and numbers of processes. The dependence of execution time, speedup, and efficiency on system parameters is investigated, allowing an assessment of the feasibility of parallel sorting for large data volumes.

Keywords: distributed system, parallel algorithm, vector sorting, Master–Worker, performance.

Вступ

Актуальність задачі сортування вектора в розподіленій системі зумовлена зростанням обсягів даних, що обробляються сучасними інформаційними системами. Послідовні алгоритми сортування є малоефективними для великих масивів даних, оскільки не дозволяють повною мірою використовувати ресурси багатоядерних і багатовузлових обчислювальних систем.

Розподілені системи забезпечують паралельне виконання обчислень на декількох вузлах, що дозволяє скоротити час обробки даних. Сортування є базовою операцією в багатьох прикладних задачах, тому використання архітектури Master–Worker для паралельного сортування вектора є доцільним і практично значущим.

Метою роботи є розробка та дослідження паралельного алгоритму сортування вектора в розподіленій системі та аналіз його ефективності.

Постановка задачі дослідження

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати особливості сортування вектора в розподілених системах;
- дослідити архітектуру Master–Worker та її застосування для паралельних обчислень;
- розробити математичну модель паралельного алгоритму сортування;
- реалізувати програму паралельного сортування вектора;
- провести тестування програми та проаналізувати результати продуктивності.

Виклад основного матеріалу

Сортування вектора є однією із базових операцій обробки даних, що широко застосовується в комп'ютерних науках, зокрема в системах керування базами даних, аналітичних платформах та алгоритмах машинного навчання. Для великих обсягів даних ефективність сортування має вирішальне значення, оскільки традиційні послідовні алгоритми з часовою складністю $O(n \cdot \log(n))$ не

дозволяють повною мірою використовувати обчислювальні ресурси сучасних багатоядерних і багатовузлових систем [1–3].

З метою підвищення швидкодії застосовуються паралельні та розподілені методи сортування, які передбачають одночасне виконання обчислень на декількох вузлах. У таких підходах вхідний вектор поділяється на частини, що обробляються незалежно, після чого результати об'єднуються у фінальний відсортований масив. Це дозволяє зменшити загальний час виконання алгоритму та підвищити масштабованість програмної реалізації [4].

Однією з найбільш поширених архітектур паралельних обчислень є архітектура Master–Worker, у якій головний процес відповідає за розподіл вхідних даних і координацію виконання, а робочі процеси здійснюють локальне сортування підвекторів [5]. Master розбиває вхідний вектор на декілька підвекторів приблизно однакового розміру та передає їх Worker-вузлам. Кожен Worker виконує локальне сортування свого фрагмента, після чого результати передаються назад до головного процесу.

Об'єднання частково відсортованих підвекторів виконується за допомогою багатошляхового злиття (k-way merge), що дозволяє ефективно сформувати глобально впорядкований вектор та зменшити кількість операцій порівняння [6]. Такий підхід забезпечує коректність результату та підвищує продуктивність алгоритму при роботі з великими обсягами даних.

Використання розподіленої архітектури забезпечує паралельну обробку даних, зменшення навантаження на окремі вузли та можливість масштабування системи шляхом додавання нових обчислювальних ресурсів. Основними напрямками оптимізації паралельного алгоритму сортування є паралелізація обчислень, ефективний розподіл ролей у моделі Master–Worker, застосування оптимізованих алгоритмів злиття та зменшення накладних витрат на синхронізацію й обмін даними [7].

Аналіз результатів тестування програми виконано для різної кількості елементів вхідного вектора та різної кількості обчислювальних процесів (табл. 1, 2).

Таблиця 1 — Результати функціонального тестування

Набір даних	Вхідні дані	Очікуваний результат	Отриманий результат	Статус
Набір 1	[5, 1, 3, 2]	[1, 2, 3, 5]	[1, 2, 3, 5]	Успішно
Набір 2	[10, -3, 7]	[-3, 7, 10]	[-3, 7, 10]	Успішно

Таблиця 2 – Результати тестування продуктивності паралельного сортування

Розмір масиву	Кількість потоків	Середній час локального сортування, с	Час глобального злиття, с	Загальний час роботи, с	Час послідовного сортування, с
500 000	8	0.11381	0.125215	0.30205	0.270626
1 000 000	15	0.26172	0.375205	0.74052	0.360214
2 000 000	8	0.07908	0.620061	1.46338	0.752688
5 000 000	15	1.69688	1.611321	3.81902	2.030291

Проаналізувавши обчислені коефіцієнти прискорення та коефіцієнти ефективності при різних розмірах вхідного вектора і різних кількості процесів, можна зробити висновки:

- при невеликих розмірах вхідних даних використання паралельного сортування є малоефективним, оскільки накладні витрати перевищують вигоду від паралелізації;
- збільшення кількості процесів при сортуванні великих вхідних векторів призводить до зростання коефіцієнта прискорення, проте супроводжується зменшенням коефіцієнта ефективності.

Таким чином, основними факторами, які впливають на час виконання програми, коефіцієнт прискорення та коефіцієнт ефективності, є розмір вхідного вектора, кількість обчислювальних процесів та особливості програмної реалізації паралельного алгоритму сортування. Зміна алгоритму

локального сортування або способу об'єднання відсортованих підвекторів може призвести до покращення або погіршення досліджуваних показників.

Висновки

У роботі досліджено задачу сортування вектора в розподіленій системі та обґрунтовано доцільність використання паралельних обчислень для підвищення продуктивності. Розглянуто архітектуру Master–Worker як ефективний підхід до організації розподіленого сортування.

Реалізовано програмний алгоритм паралельного сортування та проведено його тестування для різних розмірів вхідних даних і кількості процесів. Проаналізовано час виконання, коефіцієнти прискорення та ефективності.

Встановлено, що збільшення кількості процесів призводить до незначного зростання коефіцієнта прискорення, однак супроводжується зменшенням коефіцієнта ефективності через накладні витрати на обмін даними. Отримані результати підтверджують доцільність використання паралельного сортування для великих обсягів даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Боярчук В. П. Чисельні методи розв'язання нелінійних рівнянь. Київ: Наукова думка, 2010. 320 с.
2. Гершунська О. М. Методи та алгоритми чисельного розв'язання нелінійних систем рівнянь. Харків: Видавництво ХНУ, 2015. 400 с.
3. Кривецький О. В. Паралельні алгоритми для нелінійних задач. Вінниця: ВНТУ, 2018. 280 с.
4. Фрідман А. М. Введення в паралельні обчислення. Львів: ЛНУ ім. І. Франка, 2017. 150 с.
5. Методи оптимізації та паралельного розв'язання нелінійних рівнянь. Комп'ютерні науки. 2020. Вип. 3(21). С. 87-95.
6. Яковенко М. О. Паралельні обчислювальні системи та їх використання в наукових дослідженнях. Київ: Видавництво НТУУ, 2014. 190 с.

Очеретна Анастасія Богданівна – студентка кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: anastasiaoceretna3@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 .

Ocheretna Anastasiia Bohdanivna – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: anastasiaoceretna3@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 .