УДК 004.421:004.65

В. О. Сірак[1]
Н. В. Добровольська[1]

АЛГОРИТМИ СОРТУВАННЯ ВЕЛИКИХ ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ НА ПРИКЛАДІ MAPREDUCE

[1]Вінницький національний технічний університет

Анотація

У тезах розглянуто проблему сортування даних, обсяг яких перевищує оперативну пам'ять одного комп'ютера. Проаналізовано класичні алгоритми сортування та показано обмеження їхнього застосування в контексті Big Data. Основну увагу приділено моделі MapReduce, яка є стандартом де-факто для розподіленої обробки даних. Детально описано фазу "Shuffle and Sort" як ключовий механізм, що гарантує впорядкованість даних. Наведено таблицю порівняння підходів до сортування та зроблено висновки щодо ефективності розподіленого підходу.

Ключові слова: великі дані, MapReduce, зовнішнє сортування, Hadoop, розподілені обчислення.

Abstract

The thesis examines the problem of sorting data volumes that exceed the RAM capacity of a single machine. Classical sorting algorithms are analyzed, and the limitations of their application in the context of Big Data are shown. The focus is on the MapReduce model, which is the de facto standard for distributed data processing. The "Shuffle and Sort" phase is described in detail as a key mechanism that guarantees data ordering. A comparative table of sorting approaches is provided, and conclusions regarding the efficiency of the distributed approach are drawn.

Keywords: big data, MapReduce, external sorting, Hadoop, distributed computing.

Вступ

Сортування є однією з фундаментальних операцій в інформатиці, яка використовується в базах даних, пошукових системах та аналітичних платформах. Класичні алгоритми, такі як MergeSort або QuickSort, демонструють високу ефективність ($O(n \log n)$ в середньому) при роботі з масивами, що повністю вміщуються в оперативну пам'ять [1]. Однак, з появою феномену "великих даних" (Big Data), коли обсяги інформації сягають терабайтів і петабайтів, традиційні підходи стають неефективними через обмеження апаратного забезпечення. Виникає потреба у розподілених обчисленнях, де сортування виконується паралельно на кластерах з тисяч машин. Метою даної роботи є аналіз моделі програмування MapReduce [2] як основного методу сортування великих даних та дослідження внутрішніх механізмів, що забезпечують його ефективність.

Результати дослідження

У ході дослідження було проаналізовано архітектуру MapReduce, запропоновану компанією Google [2]. Ключовою особливістю цієї моделі є гарантія того, що вхідні дані для кожної функції Reduce відсортовані за ключем. Це досягається завдяки прихованій від програміста фазі, яка в термінології Hadoop отримала назву "Shuffle and Sort" [3].

Процес сортування в розподіленому середовищі відбувається у два етапи:

1. Сортування на стороні Map (Map-side sort). Кожен вузол, виконуючи функцію Map, не просто генерує проміжні дані, а одразу записує їх у буфер. Коли буфер заповнюється, дані сортуються за ключем та розбиваються на секції (partitions) відповідно до кількості майбутніх редукторів. Ці відсортовані дані записуються на локальний диск у тимчасові файли (spill files). Після завершення всіх Map-завдань, ці тимчасові файли зливаються в один відсортований файл [3].
2. Злиття на стороні Reduce (Reduce-side merge). Вузли, що виконують Reduce, отримують відсортовані секції з різних Map-вузлів, після чого виконують їхнє фінальне злиття, в результаті чого утворюється глобально відсортований набір даних для кожного ключа.

Такий підхід дозволяє масштабувати сортування практично лінійно зі збільшенням кількості вузлів у кластері, що є недосяжним для класичних алгоритмів. Для порівняння наведемо таблицю складності алгоритмів.

Табл. 1. Порівняння підходів до сортування

Алгоритм / Підхід	Область застосування	Часова складність	Масштабованість
QuickSort / MergeSort	Дані вміщуються в RAM одного сервера	$O(n \log n)$	Обмежена апаратним забезпеченням
Зовнішнє сортування злиттям	Дані на диску, більші за RAM	$O(n \log n)$	Обмежена одним вузлом
MapReduce (розподілене)	Великі дані (терабайти/петабайти) на кластері	$O\left(\frac{n}{p} \log \frac{n}{p}\right)$	Лінійна (при додаванні вузлів)

де p — кількість вузлів у кластері, накладні витрати на мережу не враховані.

Висновки

Класичні алгоритми сортування [1] залишаються незамінними для задач, де дані можна обробити на одній машині. Однак для обробки великих даних необхідний перехід до розподілених обчислень. Модель MapReduce [2] та її реалізація в Apache Hadoop [3] пропонують елегантне вирішення проблеми сортування на кластерах. Вбудований механізм "Shuffle and Sort" автоматично гарантує впорядкованість даних, дозволяючи розробнику зосередитись на бізнес-логіці, а не на проблемах паралелізації та розподілу даних. Таким чином, ефективність сортування в сучасних системах визначається не стільки вибором алгоритму, скільки архітектурою розподіленої системи та ефективністю фази shuffle.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms* (4th ed.). MIT Press.
2. Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. У *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI'04)*. San Francisco, CA: USENIX Association.
3. White, T. (2015). *Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale* (4th ed.). Sebastopol, CA: O'Reilly Media.

Науковий керівник: **Добровольська Наталія Вікторівна** – к.пед.н., доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: dobr_n_v@vntu.edu.ua

Сірак Владислав Олегович – студент групи 1КІ-25мс, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: vladsirak9@gmail.com

Supervisor: **Dobrovolska Nataliia V.** – Ph.D., Associate Professor of the Computer Engineering Department, Vinnytsia National Technical University, Vinnytsia, e-mail: dobr_n_v@vntu.edu.ua

Sirak Vladislav O. – student of group 1KI-25ms, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vladsirak9@gmail.com