Садовий Я.С., аспірант,
Воротніков В.В., д.т.н., доцент
Державний університет «Житомирська політехніка»

ДОСЛІДЖЕННЯ ОПТИМІЗАЦІЇ ПРОЦЕСУ ПОДІЛУ НА СЕГМЕНТИ ДЛЯ ПОБУДОВИ РОЗПОДІЛЕНИХ УЗАГАЛЬНЕНИХ СУФІКСНИХ ДЕРЕВ

Суфіксні дерева є важливими структурами даних, що застосовуються для ефективного пошуку підрядків та аналізу великих текстових масивів. Вони дозволяють виконувати такі операції, як пошук підрядків, порівняння рядків, визначення повторюваних послідовностей та інші завдання з лінійною або майже лінійною складністю [1]. Однак побудова таких структур при значних обсягах даних досі залишається складною задачею.

В умовах сучасного інформаційного середовища спостерігається постійне зростання обсягів даних. Наприклад, у галузі геноміки секвенування ДНК породжує великі масиви інформації, що потребують швидкої обробки. Подібним чином, пошукові системи висувають дедалі вищі вимоги до швидкого пошуку й аналізу текстових даних. Стандартні підходи до побудови суфіксних дерев, зокрема алгоритм Укконена [2], не завжди забезпечують достатню ефективність при обробці значних масивів інформації. Через це виникає необхідність застосування розподілених методів, які дозволяють розподілити обчислення між кількома вузлами чи процесами, що забезпечує вищу продуктивність і масштабованість.

Однією з ключових проблем при розподіленій побудові узагальнених суфіксних дерев є ефективна організація поділу даних на сегменти. Некоректний поділ може спричинити втрату інформації на межах сегментів, оскільки суфікси, що починаються на цих межах, можуть бути оброблені неправильно або зовсім втрачені. Це суттєво знижує точність та ефективність обчислень. Крім того, невдалий поділ здатний створити нерівномірне навантаження на окремі вузли кластеру, що призводить до неефективного використання ресурсів та зниження загальної продуктивності системи.

Можна виокремити три алгоритми оптимізації процесу сегментації даних для побудови розподілених узагальнених суфіксних дерев: гібридний поділ (ГП), рекурсивний поділ (РП) і динамічний адаптивний поділ (ДАП). Застосування цих алгоритмів дозволяє отримати широкий набір результатів для аналізу ефективності методів побудови суфіксних дерев на різних типах даних.

– Гібридний поділ із балансуванням навантаження поєднує декілька підходів до сегментації даних: початковий розподіл за допомогою фіксованих сегментів та подальше динамічне коригування сегментів з урахуванням реального навантаження вузлів. Це дозволяє оптимізувати розподіл даних та ефективніше використовувати обчислювальні ресурси.

– Рекурсивний поділ базується на принципі поділу даних на частини, що зберігають властивості початкової множини в меншому масштабі. Це дозволяє структуровано організовувати інформацію зі збереженням її внутрішньої структури. Основні переваги алгоритму—ітеративний підхід до сегментації та адаптивність до різних типів і структур даних.

– Динамічний адаптивний поділ уможливлює зміну розміру сегментів під час роботи, реагуючи на зміни навантаження й забезпечуючи оптимальний розподіл ресурсів.

Дослідження проводилось на п'яти різнорідних наборах даних: геномні послідовності, тексти природної мови, логи веб-серверів, програмний код, дані сенсорів. Такий вибір пояснюється різноманітністю структури та складності даних, що дозволяє всебічно оцінити ефективність алгоритмів для різних типів інформації, отримати репрезентативні результати та перевірити адаптивність алгоритмів до різних задач.

Отримані результати підтверджують, що всі три алгоритми демонструють покращення часу виконання, прискорення, ефективності використання ресурсів та точності об'єднання сегментів порівняно з послідовним алгоритмом побудови дерева. Алгоритм ДАП показав найкращі результати за всіма метриками серед досліджуваних методів. Особливо ефективними для великих наборів даних є алгоритми на основі ДАП і ГП, які значно зменшують час обчислень та підвищують ефективність аналізу. Використання архітектури Master-Worker забезпечує масштабованість системи за рахунок можливості додавання нових вузлів, підвищуючи обсяги та ефективність обробки даних, що особливо важливо для сучасних високопродуктивних систем.

Список використаних джерел:

1. Gagie, T., Navarro, G., & Prezza, N. Fully functional suffix trees and optimal text searching in BWT-runs bounded space. Journal of the ACM (JACM), 67(1), 2020.C.1-54.

2. Ukkonen, E. On-line construction of suffix trees. Algorithmica, 14(3), 1995. С. 249-260.