DOI: 10.17117/na.2016.12.02.071

<http://ucom.ru/doc/na.2016.12.02.071.pdf>

Поступила (Received): 23.12.2016

Курышов Д.М., Чумак Б.Б. Параллельный алгоритм сортировки слиянием

Kuryshov D.M., Chumak B.B.
Parallel merge sort

В статье рассмотрен пример параллельной сортировки слиянием. Показаны преимущества использования данного алгоритма

Ключевые слова: алгоритмы, сортировка, параллельные алгоритмы, сортировка слиянием

The article is dedicated to the parallel merge sort. Advantages of using the algorithm are revealed

Key words: algorithms, sorting, parallel algorithms, merge sort

Курышов Денис Михайлович

Московский технологический университет
(МИРЭА)

г. Москва, пр. Вернадского, 78

Kuryshov Denis Mikhailovich

Moscow technical university (MIREA)
Moscow, Vernadskogo ave., 78

Чумак Борис Борисович

Кандидат технических наук, доцент
Московский технологический университет
(МИРЭА)

г. Москва, пр. Вернадского, 78

Chumak Boris Borisovich

Candidate of Engineering Sciences, Associate
Professor
Moscow technical university (MIREA)
Moscow, Vernadskogo ave., 78

Алгоритмы, основанные на парадигме «Разделяй и властвуй» (Divide and conquer), легко параллельны [1]. Общий принцип распараллеливания таких алгоритмов сводиться к следующему:

- Разделить список;
- Параллельно сортировать подспски;
- Объединить сортированные списки.

Сортировка слиянием построена на принципе парадигмы «Разделяй и властвуй» [2]. Алгоритм сортировки слияния следующий:

- Если в массиве один – массив считается отсортированным, и алгоритм завершает работу;
- Если в массиве более одного элемента, массив разбивается на две части и сортируется рекурсивно.
- После сортировки двух частей массива, применяется функция слияния, которая по двум отсортированным частям получает конечный, отсортированный массив.

Сложность работы алгоритма сортировки слиянием может быть рассчитана следующим образом:

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n) = 4T\left(\frac{n}{4}\right) + 2O(n) = \dots = 2^k T(1) + kO(n);$$

$2^k = n \rightarrow k = \log n$; тогда уравнение принимает вид:

$$T(n) = nT(1) + \log n O(n);$$

так как $T(1) = \text{const}$, то $T(n) = O(n) + \log n O(n) = O(n \log n)$

Выполнение данного алгоритма можно весьма эффективно распараллелить. Так как речь идет о сравнительном анализе последовательных и параллельных вычислений, то можно допустить, что возможен запуск неограниченного количества независимых процессоров, т.е. процессоров с вычислительными ресурсами, не зависящими от других процессоров, что на практике не достижимо.

Параллельная сортировка слиянием имеет преимущества в сравнении с последовательной сортировкой слиянием. Алгоритм параллельной сортировки слиянием следующий:

- Распараллеливание обработки подзадач;
- Максимальное распараллеливание достигается с наличием одному узлу одного процессора.

Пример параллельной сортировки слиянием приведен на рисунке 1.

Рис. 1

В то время, как сортировка слиянием имеет время работы $T(n) = O(n \log n)$, параллельная сортировка слиянием будет иметь следующую сложность:

$$T(n) = T\left(\frac{n}{2}\right) + O(n) = O(\log n).$$

Список используемых источников:

1. URL: http://algotist.manual.ru/sort/bubble_sort.php
2. URL: <https://pro-prof.com/archives/1660>
3. Вирт Н. Алгоритмы и структуры данных. Невский диалект. 2008.
4. Томас Х. Кормен «Алгоритмы. Вводный курс». Вильямс. 2015.
5. Шень А. Программирование: теоремы и задачи. МЦНМО, 2014.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein – Introduction to Algorithms, MIT Press, 2009.

© 2016, Курьшов Д.М., Чумак Б.Б.

Параллельный алгоритм сортировки слиянием

© 2016, Kuryshov D.M., Chumak B.B.

Parallel merge sort