

УДК 004.75

А.Н. Мирошкин (аспірант),
И.Я. Зеленева (канд. техн. наук, доц.),
С.А. Ковалев (канд. техн. наук, доц.), **П.В. Перкин** (магістр)
Донецкий национальный технический университет
miroshkinan@gmail.com

АЛГОРИТМ РАСПРЕДЕЛЕНИЯ ЗАДАЧ ПО УЗЛАМ МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ПО КРИТЕРИЮ ВРЕМЕНИ ИХ ОБРАБОТКИ

В статье рассматриваются различные подходы к распределению задач по узлам многопроцессорной вычислительной системы. Приведен анализ классических алгоритмов, предложен новый алгоритм сбалансированного распределения задач, проведен сравнительный анализ полученных результатов.

Ключевые слова: задача, узел, время обработки, распределенные вычисления.

Введение

Обработка большого количества задач на однопроцессорной машине может занимать много времени - это зависит от производительности ресурса и параметров задач, потому в ряде случаев целесообразно для решения множества задач использовать многопроцессорную систему, что позволит уменьшить время вычислений. При этом возникает задача распределения задач по узлам вычислительного ресурса. Одним из возможных критериев является время, необходимое для обработки всего множества задач. В статье предлагается сравнение нескольких методик загрузки вычислительного ресурса [1]. Для дальнейшего изложения материала введем параметр t_i , обозначающий длительность обработки i -ой задачи на одном узле.

При выполнении последовательной обработки задач на одном узле время, затраченное на обработку всех задач, определяется как

$$T = \sum_{i=1}^N t_i, \quad (1)$$

где T – время обработки всех задач, N – количество задач. Предполагается, что обрабатываемые задачи независимы и нет необходимости обмена данными между ними (следовательно, нет и соответствующих временных затрат) [3]. Временная диаграмма последовательной обработки n задач на одном узле представлена на рис. 1.

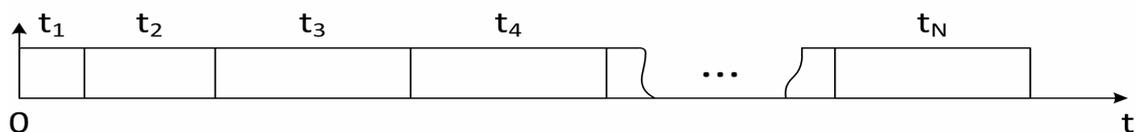


Рисунок 1 – Временная диаграмма последовательной обработки задач

Исследование эффективности классических алгоритмов распределения задач

При использовании многопроцессорного ресурса для снижения временных потерь, связанных с неравномерным использованием его узлов, необходимо выбрать методику загрузки узлов, которая удовлетворит следующему условию:

$$T \rightarrow \min \quad (2)$$

Простейший алгоритм распределения задач основывается на их сортировке в порядке возрастания времени их обработки [2]. Алгоритм заключается в следующем:

1. На каждый из K узлов распределяется по одной задаче.
2. Пока есть необработанные задачи, распределять по одной на каждый узел, который заканчивает обработку своей задачи.

Для обработки задач на нескольких узлах общее время вычислений определяется как

$$\begin{cases} T = \max_{k=1}^K (T_k); \\ T_k = \sum_{i=1}^{N_k} t_i^k, \end{cases} \quad (3)$$

где T_k – время загрузки k -ого узла; K – количество узлов; N_k – количество задач, обрабатываемых k -м узлом; t_i^k – время обработки i -й задачи на k -м узле. Будем считать, что для всех узлов время обработки одной и той же задачи одинаково.

Временная диаграмма работы процессора с двумя узлами при его загрузке задачами с возрастающими показателями t_i представлена на рис. 2.

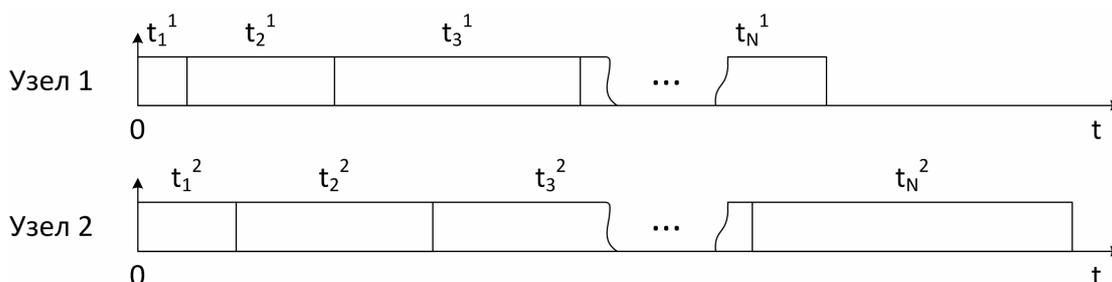


Рисунок 2 – Временная диаграмма обработки задач в порядке возрастания параметра t_i

В качестве альтернативы этому алгоритму может использоваться алгоритм распределения задач в порядке убывания времени их обработки на узле [3]. Общее время обработки задач определяется системой (3).

Временная диаграмма работы процессора с двумя узлами при его загрузке задачами с убывающими показателями t_i представлена на рис. 3.

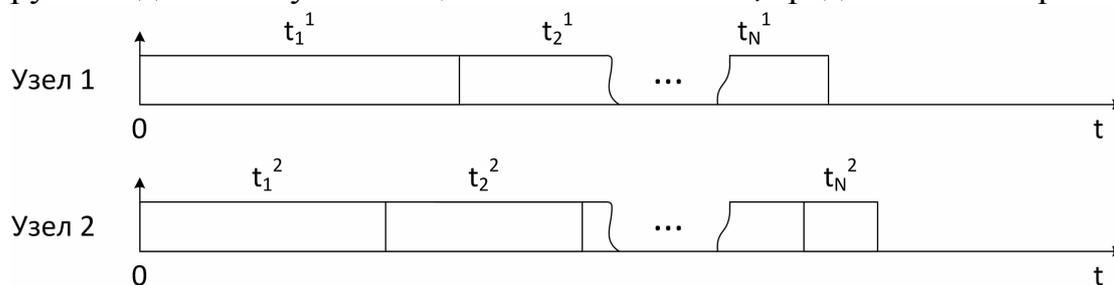


Рисунок 3 – Временная диаграмма обработки задач в порядке убывания параметра t_i

Для оценки эффективности этих двух алгоритмов был проведен ряд экспериментов. Эксперименты проводились для 4 и 40 узлов. В качестве входных данных подавались одинаковые массивы, элементы которых характеризовали время обработки задачи, содержащие от 1 до 20 задач на узел. Параметр t_i для задач формировался случайным образом в диапазоне от 1 до 100. Для каждой пары «количество узлов – количество задач» проводилось по 10 экспериментов.

В ходе экспериментов анализировалось время, необходимое для обработки одного и того же входного набора разными алгоритмами. В качестве результатов приведено отношение времени Q работы последовательного алгоритма при обратной сортировке, ко времени работы последовательного алгоритма при прямой сортировке, в зависимости от среднего количества задач на один узел n . Результат исследований приведен на рис. 4.

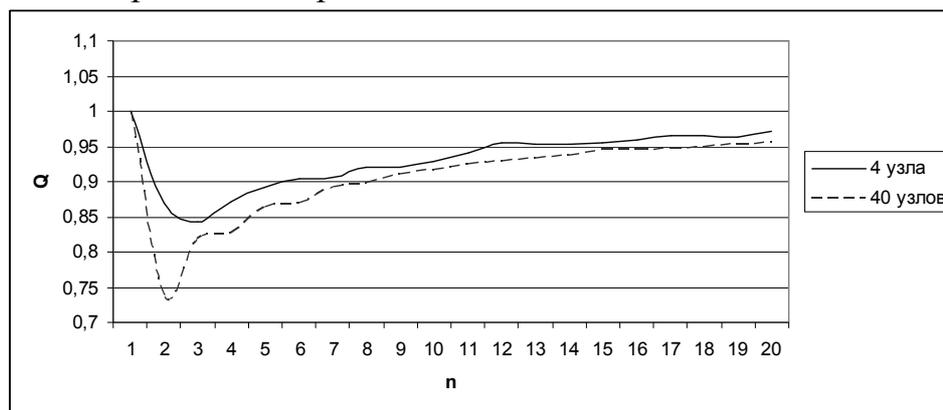


Рисунок 4 – Сравнительный анализ классических алгоритмов

Как видно из графика, при любом наборе входных данных последовательный алгоритм с сортировкой задач в порядке уменьшения параметра t_i оказался более эффективным (до 27 %).

Предлагаемый алгоритм распределения задач по узлам

В ходе проведения исследований был разработан альтернативный алгоритм, в основу которого положена балансировка нагрузки на каждый узел.

Основная идея алгоритма заключается в приближении общего времени обработки каждым узлом своих задач к параметру

$$T_{cp} = \frac{1}{K} \sum_{i=1}^N t_i . \quad (4)$$

На рис. 5 приведена графическая схема предложенного алгоритма.

Для определения эффективности алгоритма проводилось сравнение временных результатов его работы с результатами работы последовательного алгоритма сортировки задач в порядке убывания параметра t_i . Исследования выполнялись для двух, четырех и сорока узлов. Выбор обусловлен тем, что на персональном компьютере, на котором проводились исследования, установлен двухъядерный процессор (2 узла), а узлы кластера ДонНТУ Neclus оборудованы четырёхъядерными процессорами и исследования проводились для одного (4 узла) и для десяти (40 узлов) процессоров. В качестве входных данных подавались массивы, элементы которых характеризовали время обработки задач. Проводились эксперименты с различным количеством задач: от 1 до 30 задач на один узел. Параметры t_i задач выбирались случайным образом в диапазоне от 1 до 100. Для каждого набора задач и узлов проводилось по 10 экспериментов, так как временные значения в большой степени зависят от сложности задач, и такой подход дает возможность получить усредненные значения, то есть увидеть общую тенденцию. Результаты исследований приведены на рис. 6.

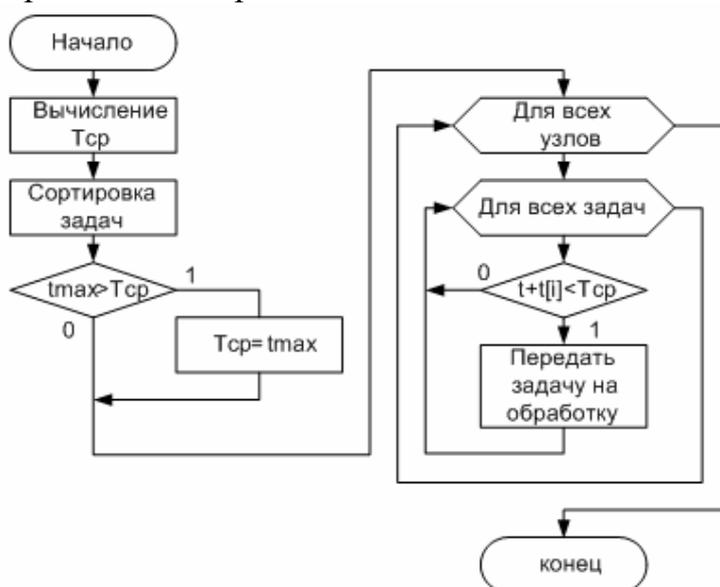


Рисунок 5 – Алгоритм сбалансированной загрузки узлов

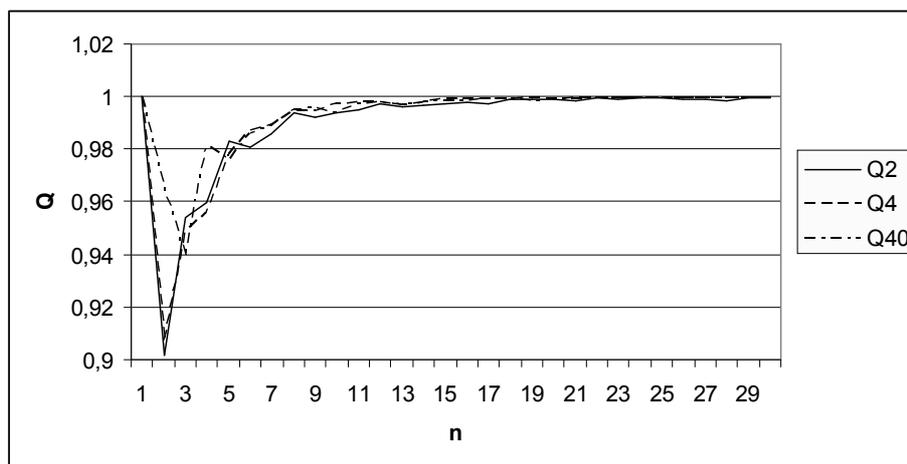


Рисунок 6 – Сравнительный график эффективности предложенного алгоритма и алгоритма обработки задач в порядке убывания параметра t_i

В ходе проведенных исследований были получены результаты, позволяющие судить об эффективности предложенного алгоритма в сравнении с классическими алгоритмами. Предложенный алгоритм сбалансированного распределения наиболее эффективен при небольшом количестве задач – до 20 задач на 1 узел.

Вывод

При использовании предложенного алгоритма более экономно используются узлы многопроцессорной вычислительной машины. Исследования для машины с четырьмя узлами показали, что в некоторых случаях один из узлов оставался свободным. Для ресурса из 40 узлов до 17 их них оставались незадействованными, что свидетельствует не только о временной эффективности алгоритма сбалансированного распределения, но и об аппаратной – свободные узлы можно использовать для решения других задач. Результатом исследований является доказательство эффективности предложенного алгоритма распределения задач по отношению к классическим алгоритмам при небольшом количестве задач на один узел.

Список литературы

1. Бройнль Томас. Паралельне програмування: Початковий курс: Навч. Посібник / Вступ. Слово А. Ройтера: Пер. з нім. В.А. Святного. – К. : Вища шк., 1997. – 358 с.\
2. Распределенные системы и алгоритмы / Миков А.И. , Замятина Е.Б. // електронний конспект лекцій. Режим доступа <http://www.intuit.ru/department/algorithms/distrsa/9>
3. Копысов С.П.. Динамическая балансировка нагрузки для параллельного распределенного метода декомпозиции области. - М.: 2003

Надійшла до редколегії 13.09.2012 р.

Рецензент: канд. тех. наук, доц. Цололо С.А.

О.М. Мірошкін, І.Я. Зеленьова, С.О. Ковальов, П.В. Перкін
Донецький національний технічний університет

Алгоритм розподілення задач по вузлам багатопроцесорної обчислювальної системи за критерієм часу їх обробки. В статті розглядаються різні прийоми розподілення задач по вузлах багатопроцесорної обчислювальної системи. Наведено аналіз класичних алгоритмів, запропоновано новий алгоритм збалансованого розподілення задач, проведено порівняльний аналіз отриманих результатів.

Ключові слова: задача, вузол, час обробки, розподілені обчислення.

A.N. Miroshkin, I.J. Zeleneva, S.A. Kovalev, P.V. Perkin
Donetsk National Technical University

Algorithm of task sharing between the nodes of multiprocessor computer system according to their processing time. Different approaches to tasks sharing between the nodes of multiprocessor computer systems are discussed. Classic algorithms are analyzed and a new algorithm for balanced sharing is proposed. Comparison and analysis of the results is done for the given algorithms.

Keywords: task, node, processing time, distributed computing.