

УДК 004.75

МЕТОДИКА РАСПРЕДЕЛЕНИЯ ЗАДАЧ ПО УЗЛАМ МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ПО КРИТЕРИЮ ДЛИТЕЛЬНОСТИ ИХ ОБРАБОТКИ

Перкин П.В., Мирошкин А.Н., Зеленева И.Я.
Донецкий национальный технический университет
perkinpv@gmail.com

В статье рассматриваются различные подходы к распределению задач по узлам многопроцессорной вычислительной системы. Приведен анализ тривиальных алгоритмов, предложен новый подход, проведен сравнительный анализ, получены результаты.

Введение

Обработка большого количества задач на однопроцессорной машине может занимать большое количество времени (зависит от производительности ресурса и трудоемкости задач), поэтому 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. Для дальнейшего изложения материала введем параметр t , обозначающий длительность обработки 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, t_i – трудоемкость i -ой задачи. Предполагается, что обрабатываемые задачи независимы и нет необходимости обмена данными между ними (следовательно, нет и соответствующих временных затрат). Временная диаграмма последовательной обработки n задач на одном узле представлена

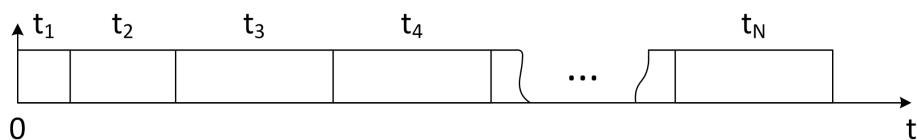


Рисунок 1. Последовательная обработка задач

на рис. 1.

Общая постановка проблемы

При использовании многопроцессорного ресурса для снижения временных потерь, связанных с неравномерным использованием его узлов, необходимо выбрать

методику загрузки узлов, которая удовлетворит следующему условию:

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

Одним из тривиальных алгоритмов загрузки является алгоритм обработки задач, отсортированных в порядке возрастания времени их обработки. 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.

Время обработки задач на нескольких узлах 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,

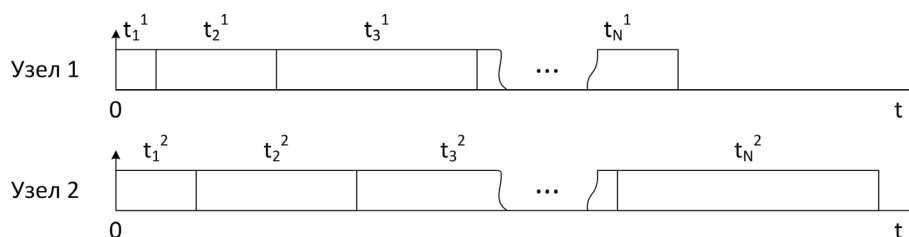


Рисунок 2. Обработка задач в порядке возрастания параметра t_i

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.

В качестве альтернативы этому алгоритму может использоваться алгоритм распределения задач в порядке убывания времени их обработки на узле. Общее время

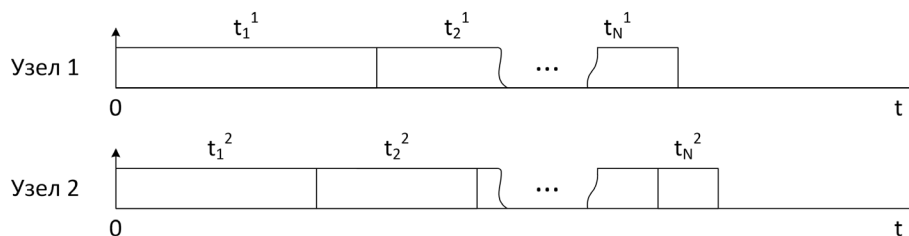


Рисунок 3. Обработка задач в порядке убывания параметра t_i

обработки задач определяется системой (2).

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

Для оценки этих двух алгоритмов был проведен ряд экспериментов. Эксперименты проводились для 4 и 40 узлов. В качестве входных данных подавались одинаковые массивы трудоемкостей задач, содержащие от 1 до 20 задач на узел. Трудоемкости

задач формировались 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

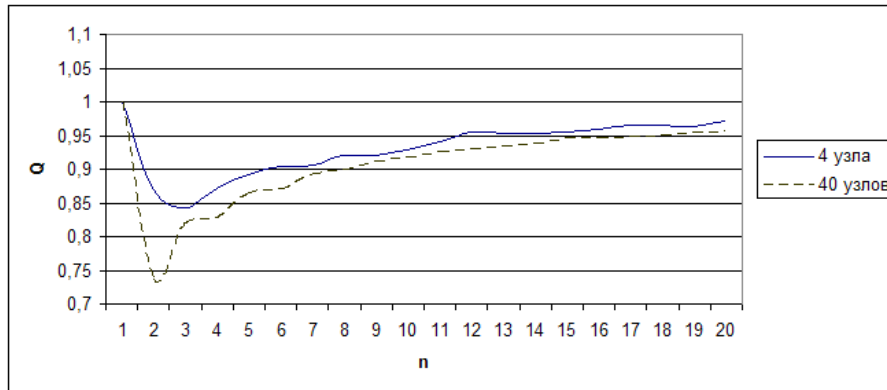


Рисунок 4. Сравнительный анализ тривиальных алгоритмов

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 приведен на рис. 4.

При любом наборе входных данных последовательный алгоритм с сортировкой задач в порядке уменьшения параметра t_i оказался более эффективным (до 27%).

Предложенная методика

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

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

$$T_{cp} = \frac{1}{K} \sum_{i=1}^N t_i. \quad (3)$$

Пошаговое описание алгоритма:

1. Вычислить параметр T_{cp} по формуле (3).
2. Отсортировать задачи в порядке уменьшения значения параметра t_i .
3. Если максимальное время выполнения одной задачи $t_{max} > T_{cp}$, $T_{cp} = t_{max}$.
4. Определить задачу-претендент на обработку для очередного свободного узла. Для задачи-претендента значение $|T_{cp} - t - t_i| \rightarrow \min$, где t – время от начала обработки всего множества задач.
5. Если все задачи распределены, переход на п. 7.
6. Ожидание завершения обработки любой из задач. Переход на п. 4.
7. Конец

Для определения эффективности алгоритма проводилось сравнение временных результатов его работы с результатами работы последовательного алгоритма с сортировкой трудоёмкости задач в обратном порядке. Исследования проводились для двух, четырех и сорока узлов. В качестве входных данных подавались массивы, элементы которых характеризовали трудоёмкость задач. Проводились эксперименты

с различным количеством задач: от 1 до 30 задач на один узел. Сложности 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

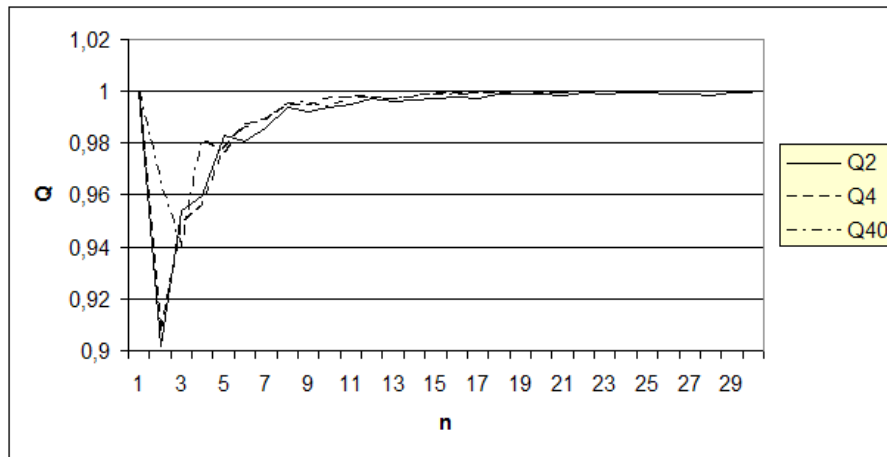


Рисунок 5 Сравнительный анализ предложенного алгоритма с алгоритмом обработки задач в порядке убывания параметра t_i

зависят от сложности задач, и такой подход дает возможность получить усредненные значения.

На рис. 5 приведены результаты исследований.

Вывод

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

Кроме того, при использовании предложенного алгоритма более экономно используются узлы многопроцессорной вычислительной машины. Исследования для машины с четырьмя узлами показали, что в некоторых случаях показывали один из узлов оставался свободным. А для ресурса из 40 узлов до 17 их них оставались незадействованными.

Список источников

- [1] Высокопроизводительные компьютеры – под. ред. Воеводина В.В. [Электронный ресурс] – Режим доступа: parallel.ru/computers/taxonomy/flynn.html
- [2] Динамическая балансировка нагрузки для параллельного распределенного метода декомпозиции области. – С.П. Копысов. М.: 2003