4. *Фомичев А.Н.* Наркомания: экономический ущерб региона в цифрах // Российское предпринимательство. – 2011. – № 2(2). – С. 181-184.

5. Министерство внутренних дел Российской Федерации. – URL: http://мвд.рф/ (дата обращения 30.09.2021).

Гончар Д.Р.

Балансировка вычислительной нагрузки при параллельной реализации решения минимаксной задачи составления расписания методом ветвей и границ

Аннотация: Рассматривается минимаксная задача построения расписания наименьшей длины без прерываний для многопроцессорной системы. Для решения данной задачи предложен параллельный алгоритм на основе метода ветвей и границ. Исследуются возможности повышения скорости расчетов при балансировке загрузки используемых процессоров.

Ключевые слова: многопроцессорная система, работы без прерываний, расписание наименьшей длины

Введение

Задачи по построению оптимальных расписаний часто встречаются при планировании производства, управлении энергетикой, подготовке и проведении испытаний сложных технических систем, работе различных систем экологического, медицинского, промышленного мониторинга и в ряде других случаях. Одним из способов ускорения расчетов при решении подобных задач является разработка параллельных алгоритмов планирования. Для алгоритмов на основе метода ветвей и границ этот вопрос весьма актуален в связи с достаточно высокой вычислительной сложностью метода.

В данной работе приведено как описание самого алгоритма, так и подходы к балансированию вычислительной нагрузки при расчетах с использованием разного числа процессоров на Межведомственном суперкомпьютерном центре РАН.

1. Постановка задачи

Пусть имеется множество работ $N = \{1, 2, \ldots, n\}$, которое необходимо выполнить с помощью m процессоров, составляющих вычислительную систему для их обработки. Длительность выполнения работы i на процессоре j равно t_{ij} ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$). В каждый момент времени каждая работа может выполняться не более чем одним процессором, а каждый процессор может выполнять не более одной работы. Переключения с одного процессора на другой и прерывания при выполнении работ не допускаются.

Расписание выполнения работ N определим как разбиение множества N на m непересекающихся подмножеств $N_1, N_2, \ldots, N_m$ ($N = \bigcup_{j=1}^{m} N_j$; $N_{j_1} \bigcap N_{j_2} = \varnothing$ при $j_1 \neq j_2$). Работы из множества N_j приписываются процессору j и выполняются на нем одна за другой в произвольном порядке. Под загруженностью процессора j ($j = 1, 2, \ldots, m$) будем понимать величину $Q_j = \sum_{i \in N_j} t_{ij}$, а $\max_{j=1, 2, \ldots, m} Q_j$ – это длина расписания. Задача заключается в построении оптимального по быстродействию расписания, т.е. расписания минимальной длины.

При решении подобных задач применяются методы случайного и исчерпывающего поиска, методы математического программирования [1], метод ветвей и границ [2], муравьиные алгоритмы, поиск с запретами, вероятностные алгоритмы, генетические алгоритмы [3], метод имитации отжига, различные эвристические алгоритмы [4, 5], алгоритмы агрегирования и др.

2. Метод ветвей и границ

Для решения вышеприведенной задачи предлагается метод ветвей и границ, основанный на результатах работы [2]. Этот метод подразумевает последовательное разбиение исходного множества решений на подмножества (ветви дерева решений) и применение оценочных процедур для определения перспективности исследования очередной ветви дерева решений. На каждом следующем шаге новые подмножества образуются в итоге разбиения некоторых подмножеств, полученных на предыдущих шагах, пока для подмножеств, соответствующих концевым вершинам дерева, решение задачи уже не требует разбиения.

В итоге указанного разбиения мы получаем множество подзадач, которые могут обрабатываться совмещено по времени.

2.1. Дерево решений

Опишем множество всех расписаний (их число равно m^n) в виде дерева решений. Корень дерева находится на нулевом уровне и соответствует множеству всех расписаний. На первом уровне находится m вершин, каждая из которых соответствует множеству всех расписаний, в которых первая работа назначена на определенный процессор. На втором уровне дерева находится m^2 вершин, каждая из которых соответствует множеству всех расписаний, в которых первые две работы назначены на один или два определенных процессора. На n-м уровне дерева расписаний находится m^n листьев, каждый из которых соответствует некоторому расписанию выполнения множества работ N.

Пусть x_k – некоторый узел уровня k дерева расписаний, $R(x_k)$ – множество всех расписаний, соответствующих этому узлу (т.е. множество расписаний, в которых работы 1, 2, …, k назначены на определенные процессоры), x_{k+1}^j – узел уровня $k+1$ ($k < n$), связанный с узлом x_k ребром, соответствующим процессору j. Наша цель – вычисление нижней и верхней оценок минимальной длины расписания на множестве $R(x_k)$. Имея эти оценки, можно применить стандартную схему метода ветвей и границ [2].

2.2. Вычисление нижней оценки

Пусть T_j (j = 1, …, m) – загруженность процессора j после назначения первых k работ (т.е. T_j – это суммарная длительность работ из числа 1, 2, …, k, назначенных на процессор j). Нижнюю оценку $L(x_k)$ минимальной длины расписания на множестве $R(x_k)$ будем вычислять следующим образом:

$L(x_k) = \max(L_1(x_k), L_2(x_k), L_3(x_k))$, где $L_1(x_k)$, $L_2(x_k)$, $L_3(x_k)$ – это нижние оценки, вычисленные тремя различными способами.

Величина $L_1(x_k)$ вычисляется как следующий максимум: $L_1(x_k) = \max\limits_{j=1,2,\ldots,m} T_j$. При хранении величины T_1, T_2, …, T_m в виде обычного массива сложность вычисления $L_1(x_k)$ составляет $\theta(m)$.

Величина $L_2(x_k)$ вычисляется как следующий максимум:

$$L_2(x_k) = \max_{i=k+1,\ldots,n} \min_{j=1,\ldots,m} (T_j + t_{ij}) \tag{1}$$

При использовании для этого двумерного массива A с элементами $a_{ij} = T_j + t_{ij}$, $i = k+1, \ldots, n$; $j = 1, 2, \ldots, m$ сложность вычисления величины $L_2(x_k)$ составляет $\theta(mn)$.

Величина $L_3(x_k)$ вычисляется по формуле

$$L_3(x_k) = \frac{1}{m}\left(\sum_{j=1}^{m} T_j + \sum_{i=k+1}^{n} \min_{j=1,\ldots,m} t_{ij}\right) \tag{2}$$

Величину $\min\limits_{j=1,\ldots,m} t_{ij}$ вычислим для всех $i = 1, 2, \ldots, n$ сразу до начала вычисления нижних оценок. Тогда сложность вычисления величины $L_3(x_k)$ составляет $O(n + m)$. Перейдем в дереве расписаний от узла x_k к узлу $x_{k+1}^{j_0}$, $k < n$ (т.е. будем считать, что работа $k+1$ назначена на процессор j_0).

Тогда

$$L_3(x_{k+1}^{j_0}) = \frac{1}{m}\left(\sum_{j=1}^{m} T_j + t_{k+1,j_0} + \sum_{i=k+1}^{n} \min_{j=1,\ldots,m} t_{ij}\right) \tag{3}$$

Вычислим разность

$$L_3(x_{k+1}^{j_0}) - L_3(x_k) = \frac{1}{m}\left(t_{k+1,j_0} - \min_{j=1,\ldots,m} t_{k+1,j}\right) \tag{4}$$

Таким образом,

$$L_3(x_{k+1}^{j_0}) = L_3(x_k) + \frac{1}{m}\left(t_{k+1,j_0} - \min_{j=1,\ldots,m} t_{k+1,j}\right) \tag{5}$$

и с помощью данного рекуррентного соотношения, используя $L_3(x_k)$, величина $L_3(x_{k+1}^{j_0})$ вычисляется за время $O(1)$.

2.3. Вычисление верхней оценки

В качестве верхней оценки $H(x_k)$ минимальной длины расписания на множестве $R(x_k)$ возьмем длину расписания, в котором работы $1, 2, \ldots, k$ в соответствии с вершиной x_k дерева расписаний распределены на процессоры, а работы $k+1, \ldots, n$ распределяются по следующему «жадному» алгоритму. Пусть уже распределены работы 1, 2, …, p ($k \le p < n$), T_j – загруженность процессора j ($j = 1, 2, \ldots, m$) и $\min(T_1 + t_{p+1,1}, \ldots, T_m + t_{p+1,m}) =$ $T_{j_0} + t_{p+1,j_0}$. Тогда работа p+1 назначается на процессор j_0.

Указанные действия повторяются для $p = k, k + 1, \ldots, n - 1$. Сложность процедуры вычисления величины $H(x_k)$ равна $O(mn)$.

2.4. Ветвление

Последовательное разбиение множества допустимых решений на подмножества происходит следующим образом: на каждом последующем шаге новые подмножества получаются в результате разбиения некоторых подмножеств, полученных на предыдущих шагах. Так образуется выше упоминавшееся дерево решения исходной задачи. Такое разбиение продолжается до тех пор, пока для подмножеств, соответствующих концевым вершинам дерева, решение задачи уже не требует разбиения.

В итоге разбиения начальная задача распадается на ряд подзадач, которые могут решаться в заметной степени вне зависимости друг от друга.

При этом представляется целесообразным поддерживать определенные связи между порожденными подзадачами, что связано с тем, что дерево решения может оказаться не вполне хорошо уравновешенным, что приводит к тому, что какая-то часть процессоров вычислительной системы оказывается загруженными неравномерно. Другая причина в том, что возникающие при попытке уравновешивания нагрузки зависимости по данным между подзадачами, связанные с передачей оценок, наилучших значений оптимизируемого функционала и других подобных сведений, могут приводить к большим накладным расходам на взаимодействие процессов, препятствующих повышению параллельной эффективности.

Для преодоления перечисленных причин снижения эффективности распараллеливания решения задачи применяется распределение обменов по вычислительному пространству, методы оптимизации загрузки процессов и минимизации обменов данными [6, 7].

Для реализации метода ветвей и границ в данной работе автором был выбран вариант метода назначаемых деревьев.

Литература:
1. *Алексеев О.Г.* Комплексное применение методов дискретной оптимизации. – М.: Наука, 1987. – 248 с.
2. *Фуругян М.Г.* Некоторые алгоритмы решения минимаксной задачи составления многопроцессорного расписания // Известия Российской академии наук. Теория и системы управления. – 2014. – № 2. – С. 50-56.
3. *Костенко В.А., Смелянский Р.Л., Трекин А.Г.* Синтез структур вычислительных систем реального времени с использованием генетических алгоритмов // Программирование. – 2000. – № 5. – С. 63-72.
4. *Brucker P.* Scheduling Algorithms. – Heidelberg: Springer, 2001. – 365 p.
5. *Гончар Д.Р.* Параллельная реализация мультиоценочного алгоритма составления многопроцессорного расписания без прерываний // Некоторые алгоритмы планирования вычислений и методы многокритериальной оптимизации для многопроцессорных систем. – М.: ВЦ РАН, 2014. – С. 21-31.
6. *Тимошевская Н.Е.* Параллельные методы обхода дерева // Математическое моделирование. – 2004. – Т. 16. № 1. – С. 105-114.
7. *Посыпкин М.А., Сигал И.Х., Галимьянова Н.Н.* Алгоритмы параллельных вычислений для решения некоторых классов задач дискретной оптимизации. – М.: ВЦ РАН, 2005. – 43 с.

Волгина О.А.

Выборочный анализ методов обработки качественной информации в количественном прогнозе

Аннотация: В работе рассматриваются выборочные методы использования качественной информации для решения широкого круга задач долгосрочного прогнозирования и анализа. Описываются свойства качественных характеристик, позволяющие им эффективно дополнять количественные методы. Особое внимание обращается на способы выделения и подготовки данных к машинной обработке.