

УДК 629.78.018

Т.С. Нікітіна

Національний аерокосмічний університет ім. М.Є. Жуковського "ХАІ", Україна

НАБЛИЖЕНИЙ МЕТОД РОЗВ'ЯЗУВАННЯ ЗАДАЧІ РОЗПОДІЛУ РЕСУРСІВ НА ОСНОВІ РІШЕННЯ ЗАДАЧІ ПРО НАЙМЕНШЕ ПОКРИТТЯ В СИСТЕМАХ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ

Запропоновано наближений метод виключень на основі рішення задачі про найменше покриття для систем паралельної обробки даних. Удосконалено метод планування завдань на основі наближеного рішення задачі про найменше покриття, що дозволяє за рахунок малої часової складності підвищити оперативність планування в 1,3 – 2,7 разів.

Ключеві слова: багатоядерний процесор, задача про найменше покриття, оперативність виконання задачі розподілу ресурсів, часова складність, лінійне програмування.

Вступ

Паралельні обчислювальні системи традиційно використовувалися в високопродуктивних обчисленнях, але останнім часом до них зріс інтерес через появу багатоядерних процесорів (БП), які є основним напрямком розвитку обчислювальних систем (рис. 1). Ефективність роботи системи на основі БП багато в чому залежить від правильного розподілу ресурсів системи: пам'яті, завдань між обчислювальними вузлами [1, 2]. Виникнення конкуренції черги завдань до деяких ресурсів БП або багатопроцесорної системи потребує вирішення оптимізаційної задачі, яка може бути зведена до вирішення задачі про найменше покриття (ЗНП) [3 – 5].

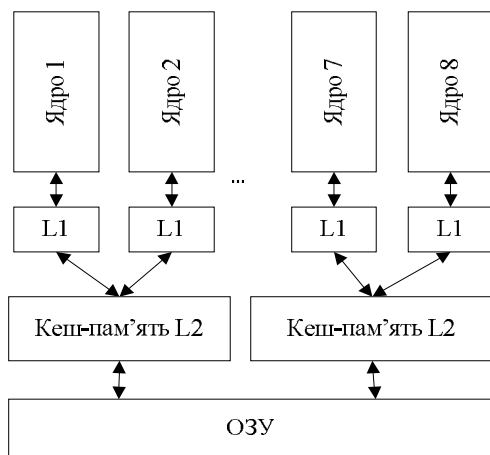


Рис. 1. Приклад архітектури БП

Метод виключень

Нехай на рішення надійшло N завдань, i потрі-

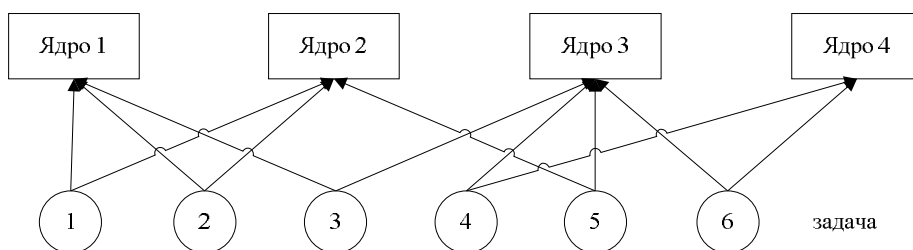


Рис. 2. Граф, що відображає можливості вирішення завдань

бно визначити мінімальну кількість ядер процесора, що забезпечує рішення всіх завдань. На рис. 2 наведені можливості щодо вирішення завдань у вигляді графа, де ядра процесора з'єднані ребрами з тими завданнями, які вони можуть виконати з необхідною ефективністю. Потрібно знайти мінімальну кількість ядер процесора, що забезпечує виконання всіх завдань з необхідною ефективністю. Двобільний граф можна представити у вигляді булевої матриці A , в якій рядкам відповідають завдання, а стовпцям ядра процесора. Елемент матриці (i, j) рівний 1, якщо j -те ядро процесора здатне з необхідною ефективністю забезпечити вирішення i -ого завдання і рівним нулю – у протилежному випадку. Тоді матриця A буде мати вигляд:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Ця задача може бути сформульована як задача лінійного булевого програмування [6], постановка якої має вигляд:

$$G = \sum_{i=1}^n x_j \rightarrow \min \quad (1)$$

при обмеженнях:

$$\sum_{i=1}^n a_{ij}x_j \geq 1; x_j \in \{0,1\}; \quad (2)$$

де
 $a_{ij} = \begin{cases} 1 - \text{якщо } i - \text{та змінна покривається } x_j; \\ 0 - \text{інаше.} \end{cases}$

В якості методу вирішення ЗНП пропонується наступна процедура планування.

Метод виключень. Нехай є M ядер процесора, при цьому кожне m -е ядро процесора в змозі вирішити деяку підмножину G_m завдань з необхідною ефективністю, де $m \in 1...M$. Припустимо, що на рішення надійшло N завдань і потрібно визначити мінімальну кількість ядер процесора, що забезпечує вирішення всіх N завдань і, як правило $N > M$. В якості методу вирішення ЗНП пропонується наступна процедура планування. Для вирішення ЗНП довільна матриця представляється у вигляді булевої функції, на основі подання булевої матриці з M стовпцями і N рядками [7].

Стовпці задаються вектором $X = \{x_1, x_2, \dots, x_M\}$, рядки – $Y = \{y_1, y_2, \dots, y_N\}$. Покриттям Q рядків Y називається така безліч стовпців, яка покриває одиницями всі рядки Y . Для визначення всіх покриттів матриці використовується алгебраїчний метод отримання по імплікантній таблиці наведених систем простих імплікант булевих функцій. Якщо кожен стовпець із сукупності $X = \{x_1, x_2, \dots, x_M\}$ розглядати як «просту імпліканту», яка покриває сукупність рядків $Y = \{y_1, y_2, \dots, y_N\}$, кожен рядок – y_m як набір змінних, що покриваються простими імплікантами, то матрицю можна представити як імплікантну таблицю булевої функції. При такій інтерпретації матриці для кожного рядка y_m можна записати диз'юнкцію стовпців, що покривають рядки, в такому вигляді:

$$\begin{aligned} dy_m &= (x_1 \vee x_k \vee \dots) \\ &\dots\dots\dots \\ dy_m &= (x_p \vee x_t \vee \dots) \end{aligned} \quad (3)$$

Кон'юнкція диз'юнкцій (3) по всіх рядках y_1, y_2, \dots, y_N матриці A утворює кон'юнктивне подання матриці A , що містить в собі всі покриття сукупності рядків Y :

$$k(Y) = dy_1 \cdot dy_2 \cdot \dots \cdot dy_r = (x_1 \vee x_k \vee \dots) \cdot \dots \cdot (x_p \vee x_t \vee \dots)$$

Розкриваючи дужки відповідно до законів дистрибутивності, виходить диз'юнктивне подання матриці A , що створює перелік всіх можливих покриттів сукупності рядків $Y = \{y_1, y_2, \dots, y_N\}$. Таким чином, довільна булева матриця представляється у вигляді булевої функції, заданої в кон'юнктивній нормальній формі, в якій число диз'юнктивів дорівнює числу рядків в матриці, а число змінних в кожному із диз'юнктивів – числу одиниць в рядку матриці A . Для визначення мінімального покриття введемо та-

ку характеристику булевої функції $f(X_1, X_2, \dots, X_M)$ у вигляді вектора

$h_q = (h_{m=1}^{i_1}, h_{m=2}^{i_2}, \dots, h_{m=M}^{i_M})_q$, де вага i_m у векторі

h_q вказує, як часто змінна зустрічається в диз'юнктах функції $f(X_1, X_2, \dots, X_M)$. Розглянемо алгоритм вирішення цієї задачі у вигляді такої процедури, яку назвемо методом виключень (МВ), для перетворення $f(X_1, X_2, \dots, X_M)$.

Алгоритм реалізації МВ.

Крок 1. У булевій функції $f(X_1, X_2, \dots, X_M)$, яка є диз'юнктивним поданням матриці A , шукається змінна X_m , яка найчастіше зустрічається в диз'юнктах, тобто змінна з максимальним значенням ваги m_i у векторі h_q .

Крок 2. Обрана змінна X_m множиться на функцію $f(X_1, X_2, \dots, X_M)$ при цьому всі диз'юнкти цієї функції, що містять змінну X_m виключаються з подальшого аналізу.

Крок 3. Якщо всі диз'юнкти виключені з аналізу та отриманий співмножник $X_p \cdot \dots \cdot X_q$ відповідає мінімальному числу стовпців, що покриває одиницями всі рядки, то алгоритм закінчує роботу, інакше перехід до кроку 2.

МВ дозволяє вирішити задачу розподілу завдань по ядрам процесора для систем на основі БП, також він може бути застосовано для багатопроцесорної системи.

Експериментальне дослідження МВ

Для оцінки ефективності МВ в роботі проведено імітаційне моделювання. Головним показником ефективності роботи алгоритму є відрізок часу, що витрачаються алгоритмом для вирішення поставленого завдання, в даному випадку необхідно визначити час необхідний на сервісне обслуговування черги завдань. Однак такий підхід не однозначно визначає ефективність алгоритмів, тому що цей показник залежить від типу обчислювальної системи, на якій виконується алгоритм, від вибору мов програмування і т.д. Тому, крім часу рішення, використовуються інші критерії оцінки ефективності алгоритмів. При аналізі алгоритмів зазвичай цікавить швидкість росту складності алгоритму, яку можна оцінити згори, використовуючи вираз типу "вимагає часу $O(n^2)$ ".

В якості основних характеристик наближених алгоритмів також використовують абсолютну δf і відносну Δf похибки отриманого наближеного рішення, які визначаються із співвідношень

$$\delta f = \left| f(x) - f(x^*) \right|, \Delta f = \left| f(x) - f(x^*) \right| / f(x^*),$$

де f – цільова функція, визначена на деякій множині M ; x – наближене рішення задачі; x^* – оптимальне рішення.

Таким чином, в якості основних критеріїв ефективності алгоритмів використовувались кількість елементарних операцій, час рішення і відносна похибка алгоритму. Імітаційне моделювання широко застосовується в дослідженнях проблеми розподілу ресурсів. Мета проведення експерименту порівняння ефективності МВ і класичних методів планування. Проведений за допомогою імітаційної моделі експеримент являє собою спостереження за поведінкою методів під впливом вхідних впливів.

На оперативність розподілу завдань в системі можуть впливати такі фактори (рис. 3): кількість завдань; кількість процесорів; розмір буфера; відсоток завдань розв'язуваних на одному ресурсі.

Вхідні, вихідні змінні і структурні допущення представлені на схемі експериментального дослідження (рис. 3). Кожен сеанс моделювання здійснює формування потоку вхідних даних, де кількість завдань, процесорів задається в інтерактивному режимі, і матриця задається за рівномірним законом розподілу. Після X циклів можна оцінити ефективність МВ.

Аналіз результатів моделювання

Визначення залежності часу рішення задачі планування від кількості вхідних завдань. Було проведено 10 сеансів моделювання з кількістю ядер процесора від 2 до 16. Наведемо приклад розподілу 150-350 завдань на 8-х ядрах процесора з розміром буфера 3 (рис. 4). Планування здійснювалося за допомогою двох методів: FIFO і МВ.

Після проведеного експерименту можна зробити висновок, що раціональним є застосування МВ для великої розмірності задач в діапазоні 150-300 (рис. 4). Були отримані наступні результати:

час розподілу потоку завдань зменшилася на 1,5-2,7 разів;

кількість вільних ядер процесора збільшилася в 1,1 разів.

Експериментальна оцінка ймовірності виконання завдання планування за обмеженої час. В комп'ютерних системах на час обслуговування запитів накладається часове обмеження, яке не повинно перевищувати деяке значення T_d [8]. Тому рішення задачі

обслуговування запитів має здійснюватися з необхідною оперативністю. Оперативність виконання задачі, що не перевищує допустимий $P(T)$, і кількісно оцінюється ймовірністю $P(T)$ обробки необхідного потоку завдань за час T :

$$P(T) = 1 - e^{-T_d/T}, \quad (4)$$

де T_d – допустимий час, за який повинна бути виконана обробка системи завдань планувальником і T – час, за який отримано рішення на основі МВ. Важливо щоб часова складність алгоритму не перевищувала $T_d = 10 \dots 100$ мс для планування завдань в режимі реального часу. Значення показника оперативності істотно залежить від розмірності задачі.

Аналіз результатів моделювання показує, що МВ дозволяє досягати оперативності на рівні $P(T) > 0.9$ - для $M=100$ і $N < 60$ при $T_d = 10$ мс, $N < 45$ при $T_d = 5$ мс, $N < 25$ при $T_d = 3$ мс; для $N=300$ і $M < 15$ при $T_d = 10$ мс, $M < 12$ при $T_d = 5$ мс.

Висновки

Удосконалено метод планування завдань на основі наближеного рішення задачі про найменше покриття, що дозволяє за рахунок малої часової складності підвищити оперативність планування в 1,3 – 2,7 разів. МВ дозволяє застосовувати його, як інструмент планування розподілом ресурсів в системах на базі БП, де процес планування здійснюється в режимі РВ при розмірності задачі ($n < 300$, $m < 60$), при цьому його застосування дозволяє отримувати своєчасний результат при рівні показника оперативності $P(T) > 0.9$. Експериментально доведено, що похибка запропонованого алгоритму не перевищує 4-5% і зменшується зі збільшенням розмірності задачі.

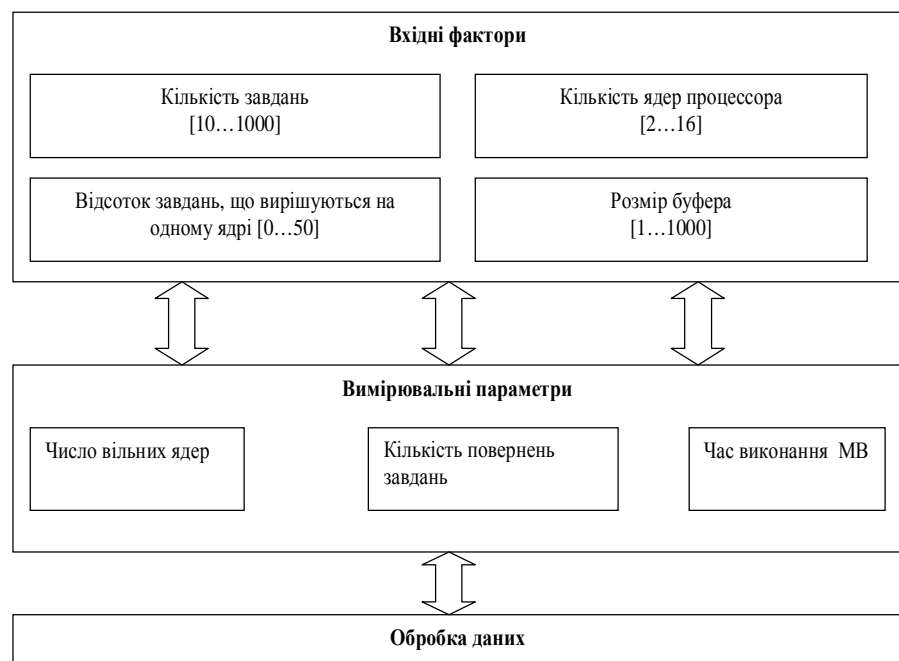


Рис. 3. Схема експериментального дослідження

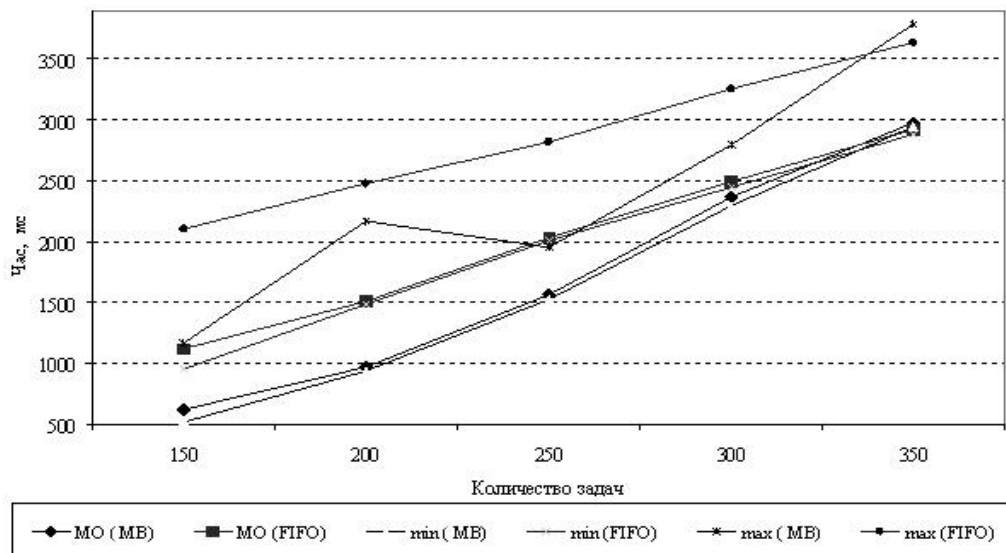


Рис. 4. Залежність часу розподілу завдань від їх кількості (ядер процесора 8, розмір буфера 3)

Список літератури

1. Levy M. *Measuring Multicore Performance. Embedded computing / M. Levy // Computer. – 2008. – № 41. – P. 99-102.*
2. Viren K. *Optimal Multicore Scheduling: An Application of ASP Techniques [Електронний ресурс] / K. Viren, J. Delgrande. – Режим доступа: <http://www.springerlink.com/content/1672mn5n19788136/>*
3. Никитина Т.С. *Метод распределения ресурсов в гетерогенных многопроцессорных системах на основе решения задачи о наименьшем покрытии / Т.С. Никитина // Міжнародна НТК «Інформаційні технології в навігації і управлінні: стан та перспективи розвитку»: Тези доповідей. – К., 2010. – С. 42.*
4. Листровой С.В. *Точный алгоритм решения задачи о наименьшем покрытии / С.В. Листровой, А.Ю. Гуль // Информатика. Институт проблем моделирования в энергетике НАНУ. – К.: Наук. думка. – 1998 – № 5. – С. 32-36.*
5. Пономаренко В.С. *Метод решения задачи о минимальном покрытии как средство планирования в GRID / В.С. Пономаренко, С.В. Листровой // Проблемы управления, РАН. – 2008 – №3. – С.78-84.*

6. Третьяк В.Ф. *Структура систолического вычислителя для приближенного решения задачи булевого линейного программирования / В.Ф. Третьяк, С.В. Кавун, А.Ю. Гуль // Материалы НТК "Обработка информации и обеспечение надежности систем управления". – Х.: НАНУ, ПАНИ, ХВУ. 1996. – С. 7.*

7. Burns A. *Scheduling Hard Real Time Systems / A. Burns // Software Engineering Journal. – 1991. – № 6(3). – P. 116 – 128.*

8. Туркин И.Б. *Анализ алгоритмов планирования задач реального времени для многопроцессорных систем / И.Б. Туркин, Т.С. Никитина // Радиоэлектронные и компьютерные системы: научно-технический журнал. – 2008. – № 6. – С. 37-41.*

Надійшла до редколегії 1.06.2011

Рецензент: д-р техн. наук, проф. В.М. Варганян, Національний аерокосмічний університет ім. М.С. Жуковського «ХАІ», Харків.

ПРИБЛИЖЕННЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧИ О НАИМЕНЬШЕМ ПОКРЫТИИ В СИСТЕМАХ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

Т.С. Никитина

Предложен приближенный метод исключений на основе решения задачи о наименьшем покрытии. Метод исключений работает на порядок быстрее, чем известные лучшие методы, что позволяет применять его, как инструмент планирования распределением ресурсов в системах, где процесс планирования осуществляется в режиме реального времени при размерности задачи ($N < 300$, $M < 60$), при этом его применение позволяет получать своевременный результат при уровне показателя оперативности $p(T) > 0.9$. Экспериментально доказано, что погрешность предложенного алгоритма не превышает 4-5% и уменьшается с увеличением размерности решаемой задачи.

Ключевые слова: многоядерный процессор, задача о наименьшем покрытии, оперативность выполнения задачи распределения ресурсов, временная сложность, линейное программирование.

APPROXIMATE METHOD OF SOLVING RESOURCE ALLOCATION PROBLEM BASED ON SOLUTIONS OF MINIMAL COVERAGE IN THE SYSTEM OF PARALLEL DATA PROCESSING

T.S. Nikitina

An approximate method of exceptions on the basis of solving the problem of minimum coverage. Method exceptions works much faster than the known best practices, it can be used as a planning tool for resource allocation in systems where the planning is done in real time with the problem dimension ($N < 300$, $M < 60$). At the same time it allows to use in getting timely results at a level of efficiency index $p(T) > 0.9$. Experimentally proved that the error of the algorithm does not exceed 4-5% and decreases with increasing dimension of the problem being solved.

Keywords: multicore processor, the problem of the minimal coverage, timeliness of the task resource allocation, time complexity, linear programming.