

Практична цінність. Наукові результати, отримані в цій статті, дають змогу вести подальші практичні дослідження по методах класифікації для усунення невизначеності.

1. Кравець Р.Б. Організація багатовимірного подання та аналізу інформації у реляційній базі даних // Вісник НУ "Львівська політехніка". 2003. № 489. 2. Netz A., Chaudhuri S. *Integration of Data Mining and Relational Databases. Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000.* 3. G. Grsrfe, U. Fayyad. *On the Efficient Gathering of Sufficient Statistics for Classification from Large SQL Databases.* – 1998. – www.aaai.org. 4. Huhtala Y., Karkainen J. *Tane: An Efficient Algorithm for discovering Functional and Approximate Dependencies*// *The Computer Journal.* 1999. – Vol. 42. – № 2. 5. Дюбуа А., Прад А. Теория вероятностей. Приложение к представлению знаний в информатике. М., 1990. 6. Мейер Д. Теория реляционных баз данных. – М., 1987. 7. Шаховська Н.Б. Застосування апарату багатозначно. Логіки у системах баз даних // Вісник НУ "Львівська політехніка". 2001. № 438. 8. Шаховська Н.Б. Методи усунення невизначеностей у базах знань, побудованих на основі реляційного підходу // Вісник НУ "Львівська політехніка". – 2001. – № 438.

УДК 004.272.44

І.Д. Яковлєва, І.Д. Лісовенко

Чернівецький національний університет імені Юрія Федьковича
кафедра комп'ютерних систем та мереж

ПІДХІД ДО ПОБУДОВИ ПОТОКОВОГО ГРАФА АЛГОРИТМУ

© Яковлєва І.Д., Лісовенко І.Д., 2005

Розглянуто метод визначення готовності даних з використанням концепції машини потоків даних та запропоновано підхід до побудови потокового графа алгоритму.

In this paper the method of definition of data readiness based on the dataflow machine conception is considered and the approach to construction of algorithm flowgraph is offered.

Вступ. Для послідовних комп'ютерів основна модель виконання – відома модель фон Неймана, яка складається з послідовного процесу, що виконується в лінійному адресному просторі. Частка доступного паралелізму в ній порівняно мала. Принцип машини потоків даних (МПД), який подано у [1, 2], є одним з архітектурних шляхів досягнення високої продуктивності обчислень для максимального використання паралелізму.

Представлення алгоритмів у формі стандартного математичного запису не дає змоги достатньо повно оцінити можливості розпаралелювання та знайти способи їхньої ефективної реалізації. Як зазначено в [3], однією з форм представлення алгоритму, яка дає змогу оцінити його обчислювальні і структурні характеристики, є зображення алгоритму у вигляді функціонального графа.

Модель обчислювальної системи, що заснована на принципі потоку даних, автоматично використовує паралелізм, притаманний розв'язуванню задачам. Це значно полегшує організацію паралельної роботи, розподіл завдань при програмуванні і організацію обміну інформацією. В результаті продуктивність обчислювальної системи істотно зростає.

Мета цієї роботи – створити метод, який визначає готовність даних в послідовній програмі з використанням концепції МПД та побудувати потокову форму графа готовності даних алгоритму, яка описана в [5].

Описання алгоритму визначення готовності даних з використанням концепції МПД та побудова ярусно-паралельної форми алгоритму. В МПД команда оголошується готовою до

виконання, як тільки обчислені всі операнди, що необхідні для її виконання. Команда, для якої ця умова виконана, є командою, що спрацювала. Ефект спрацювання команд призводить до поглинання вхідних значень і генерування вихідних [1]. Відповідно до описаного вище правила ця модель є асинхронною. Оскільки доступність обчислених операндів уможливорює одночасне виконання декількох операторів, паралельність дії є внутрішньою властивістю схем потоків даних.

Машина потоків даних не має справи з адресами комірок пам'яті, в яких містяться деякі величини. МПД працює тільки з токенами, кожен з яких становить самостійний об'єкт. Обчислення в МПД організується в такий спосіб: токени передаються між акторами, коли актор одержує всі необхідні значення для виконання операції він активізується, поглинає вхідні токени, а після закінчення операції породжує токен результату, що пересилається до іншого актору, для якого він стає вхідним.

Тому нам необхідно знайти метод, який дасть змогу в послідовній програмі визначити готовність даних для подальшого розпаралелювання програм з використанням концепції МПД.

Запропонований такий алгоритм роботи системи:

1) Нумерація операндів програми і побудова матриці залежності $S[m,m]$ операндів один від одного, де m – кількість змінних (Алгоритм 1).

2) Побудова вектора готовності даних $FR[m]$. (Алгоритм 2)

3) Перетворення матриці залежності $S[m,m]$ та вектора готовності даних $FR(m)$ за алгоритмом 3 дає матрицю готових до виконання операндів $S'[m,m]$ і вектор готовності даних на i -му етапі.

4) Аналіз $FR(m)$ і побудова ярусу потокового графа відповідно до готовності даних. Якщо $\bigwedge_{i=0}^m FR[i] = 1$, то всі дані вже готові і побудова потокового графа закінчена, інакше переходимо до п.3

Виявити паралелізм і навіть керувати ним, забезпечуючи тим самим можливість знаходження компромісних просторово-часових співвідношень дає змогу зображення графа алгоритму в потоковій формі, яке є найнаочнішою формою представлення паралельних алгоритмів, і забезпечує аналіз різних структур та дає змогу побачити просторово-часові паралелі.

Побудова матриці залежності. Перевага застосування матриць суміжностей полягає у зручності представлення, опрацювання та зберігання в комп'ютері залежностей з великою кількістю елементів. Такі матриці в комп'ютерах записують як масиви або списки зв'язності. Елементи матриць суміжностей набувають тільки два значення, отже, над ними зручно застосовувати менш трудомісткі операції алгебри логіки, ніж звичайні алгебраїчні операції.

Недоліком матричних способів задання взаємозв'язків є низька наочність. Тому для виведення результатів аналізу взаємозв'язків необхідно передбачити перетворення матриць у графи.

Алгоритм 1

Побудова матриці S :

1) Визначення кількості m операндів програми.

2) Матриця $S[m,m]$ – квадратна матриця, що задає взаємозв'язки операндів програми, де $\forall S_{ij} \in \{0,1\}$, $i = \overline{1,m}$, $j = \overline{1,m}$.

3) Кожен $S[i,j]$ - елемент матриці асоціюється з 1 у випадку, коли існує залежність між i -м та j -м операндами і 0, якщо залежність відсутня.

Побудова вектора готовності даних.

Алгоритм 2

Кроки побудови вектора FR готовності даних, де $\forall FR \in \{0,1\}$, $i = \overline{1,m}$:

1) Початкове значення $i = 0$.

2) j-й елемент вектора готовності операндів FR будують за правилом: $FR[j] = 1$, якщо

$$\sum_{j=0}^m S[i, j] = 0 \equiv \text{NOT}(\text{ORS}_{j=0}^m S[i, j]) = 0 \text{ -- всі елементи цього рядка дорівнюють нулю, тобто вже не}$$

залежать від інших операндів, отже, операнд готовий до виконання і $FR[j] = 0$, коли $\sum_{j=0}^m S[i, j] > 0$

$$\equiv \text{NOT}(\text{ORS}_{j=0}^m S[i, j]) = 1, \text{ коли хоча б один елемент } S[i, j] = 1.$$

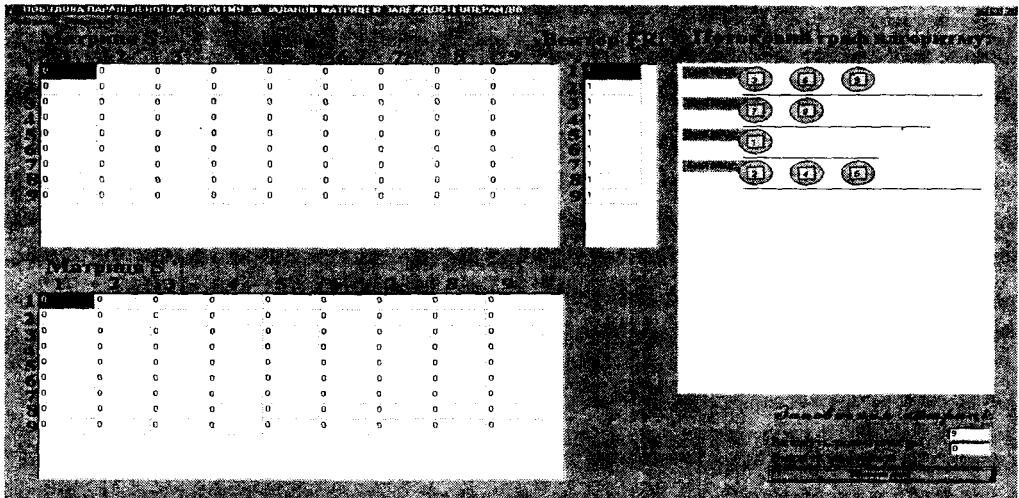
3) Інкремент i. Перехід на крок 1.

Побудова матриці готових до виконання операндів.

Алгоритм 3

Матрицю готових до виконання операндів S' , де $\forall S'_{ij} \in \{0,1\}$, $i = \overline{1,m}$, $j = \overline{1,m}$, отримують у результаті перетворень матриці залежності операндів S і вектора готовності операндів FR за правилом: $S'[i, j] = S[i, j] \text{AND} \text{NOT}(FR[j])$.

Після виконання перетворень над матрицею S та вектором FR отриманий граф алгоритму в потоковій формі. Це означає розподіл всіх його вершин по ярусах так, що в i-му ярусі розміщені тільки операнди, які залежать хоча б від одного операнду (i-1)-го ярусу і не залежать від операндів, які не входять в яруси з меншими, ніж i номерами. Всередині ярусу операнди між собою не мають з'єднань.



Приклад головного вікна програми в режимі виведення готових до виконання операндів

Рисунок зображує описані алгоритми на прикладі заданої матриці взаємозв'язків, де $m=5$. В результаті виконаної роботи ми отримали набір правил для формування масиву даних, готових до виконання.

Висновки. Запропонований в цій статті метод дає змогу визначити готові до виконання дані вхідної послідовної програми, що дає змогу побудови потокового графа алгоритму. Застосування

матриць суміжності та вектора готовності є зручним для машинного представлення, опрацювання та зберігання. Через низьку наочність матричного способу задання взаємозв'язків передбачено перетворення матриць у графи. Розглянуті алгоритми виконані як програмний модуль і реалізовані в середовищі Delphi 7.0.

1. Dennis J.B. *First version of a data-flow procedure language* // *Lect. Notes Comput. Sc.* - 1974. – Vol.19. – P.362–376. 2. Arthur H.Veen. *Dataflow Machine Architecture* // *Computing Surveys*. – 1986. – Vol.18, No.4. 3. Кузьмук В.В. *Методика алгоритмического описания и моделирования параллельных процессов управления*. – К. 1981. 4. *Системы параллельной обработки* / Под ред. Д.Ивенса. – М., 1985. 5. Сир. Ж.-К. *Метод потока операндов в многопроцессорных системах типа MIMD* // *Системы параллельной обработки*. – М., 1985. – С. 240–251.