

Аспірант Жабіна В.В.

Національний технічний університет України  
«Київський політехнічний інститут»

**МЕТОД ПАРАЛЕЛЬНОГО ФОРМУВАННЯ  
КОМАНД В ПОТОКОВИХ СИСТЕМАХ**

**Abstract**

*Valentina V. Zhabina, post-graduate student*

***Method for parallel instruction compilation in data-flow systems***

*Method of computation acceleration in data-flow systems for realization of algorithms with branches and loops is proposed. It is based on dynamic parallel instruction flows compilation. There is shown a possibility to automatically identify actor and data words during compilation on the base of task graph for data-flow systems with several instruction formation units.*

**Вступ**

При розв'язуванні задач керування і моделювання в реальному часі виникає необхідність реалізації обчислювальних алгоритмів з дрібнозернистою структурою. Прикладом таких алгоритмів є алгоритми інтерполяції функцій, розрахунку траєкторії об'єктів у багатовимірному просторі тощо. Прискорення обчислень у цьому випадку можна досягти шляхом розпаралелювання операцій. Час реалізації алгоритмів багато в чому визначається ефективністю розпаралелювання.

Більшість із сучасних технологій паралельного програмування належать до засобів статичного розпаралелювання процесів [1]. При статичному аналізі алгоритмів на етапі розробки програм не завжди вдається виявити паралельні гілки, тобто прихований паралелізм, що пояснюється недостатньою інформацією про динаміку процесів у системі.

Для усунення недоліків статичного планування використовують засоби динамічного розпаралелювання обчислень. Одним з підходів для динамічного розподілу операцій між обчислювальними вузлами є використання моделі обчислень, керованих потоком даних (потоккової моделі). Розподіл операцій між обчислювальними вузлами в цьому випадку може бути реалізований автоматично в процесі обчислень на апаратному або мікропрограмному рівні.

До одних з перших наукових праць в галузі організації обчислювальних систем, керованих потоком даних (потоківих систем), можна віднести праці [2]-[5]. Маючи структурні особливості, такі системи використовують загальну модель обчислень. Потоківі системи містять кілька обчислювальних модулів (ОМ) і одне середовище формування команд (СФК), зв'язаних комунікаційними засобами. Для реалізації даної моделі обчислень можуть бути використані ПЛІС, які містять обчислювальні ядра, модулі пам'яті і засоби комунікації. При великій частоті тактування сучасних ПЛІС основну затримку обчислень вносить загальна СФК, побудована на основі пам'яті.

Просте дублювання СФК призводить до недоліків статичних методів, тобто до ручного розпаралелювання процесів. Даним і акторам необхідно попередньо присвоїти ідентифікатори, що визначають їх приналежність до певних СФК.

У праці [6] запропоновано метод автоматичної ідентифікації об'єктів для систем з кількома СФК. Однак цей метод не може бути використаний для алгоритмів з циклами, наприклад, для ітераційних обчислень.

### **Постановка задачі**

Задачею дослідження є розроблення методу формування паралельних потоків команд у потоківих системах для підвищення ефективності паралельних обчислень при реалізації алгоритмів з циклами.

### **Потокова модель обчислень**

У системах, керованих потоком даних, команди виконуються не в заданій програмі послідовності, а за наявності готових даних (наявності всіх операндів), тобто визначальним в цьому випадку є не порядок виконання команд, а доступність даних для команди. Підготовка обчислень здійснюється на основі графа, кожній  $i$ -й вершині якого відповідає операція (функція), а кожній дузі – дані.

Операція для  $i$ -ої вершини графа описується інформаційним словом, що називають актором (actor). Актор описується кортежем

$$A_i = \langle I_i, F_i, N_i, T_i \rangle, \quad (1)$$

де  $I_i$  – ідентифікатор (унікальне ім'я) даного актора;  $F_i$  – функція перетворення даних (код операції);  $N_i$  – ім'я актора, для якого  $i$ -й актор підготовляє операнд, а  $T_i$  – сукупність ознак цього операнда.

Актори пов'язані між собою тільки за даними. Кожній дузі графа відповідає слово даних

$$D_i = \langle I_i, Q_i, N_i, T_i \rangle, \quad (2)$$

де  $Q_i$  – значення операнда.

З відповідних елементів  $A_i$  і  $D_i$  в СФК формується команда, що виконується у вільному ОМ або записується в чергу.

### Формування паралельних потоків команд

Система може мати довільну кількість ОМ. Кількість СФК має бути рівною  $k=2^j$ , де  $j=1,2,3\dots$

Формування акторів і даних здійснюється на основі графа. Модифікуємо формати об'єктів (1) і (2).

Формат акторів набуває вигляду

$$A_i = \langle M_i, I_i, F_i, K_i, L_i, T_i \rangle, \quad (3)$$

де  $M_i$  – ідентифікатор СФК (номер потоку команд);  $I_i$  – унікальне ім'я даного актора;  $F_i$  – функція перетворення даних;  $K_i$  – кількість акторів, для яких  $i$ -й актор підготовляє операнд;  $L_i = \langle I_j, M_j \rangle$  – список імен акторів  $I_j$  з ідентифікаторами  $M_j$ , для яких  $i$ -й актор підготовляє операнд;  $T_i$  – тип даних.

Дані запишемо у вигляді

$$D_i = \langle Q_i, N_i, L_i, T_i \rangle, \quad (4)$$

де  $Q_i$  – значення операнда;  $N_i$  – кількість акторів, для яких передається даний операнд.

Для формування потоків команд, тобто знаходження значень ідентифікаторів  $M_i$ , пропонується наступний алгоритм.

1. На основі вихідного графа побудувати матрицю суміжності, що показує зв'язок між акторами за даними. У якості номерів рядків і стовпців взяти імена акторів  $I_i$  або їх індекси  $i$ . Першому потоку присвоїти ідентифікатор  $M_0$ .

2. Обнулити основну діагональ отриманої матриці.

3. Для формування паралельних потоків команд розпочати циклічний перегляд нової матриці суміжності. У кожному циклі від першого до останнього рядка виконувати:

а) якщо поточний рядок не містить одиничних елементів, перейти до наступного вихідного рядка;

б) якщо рядок з індексом  $i$  містить одиничні елементи, то в якості першого актора в потоці команд взяти актор з ім'ям  $I_i$ ;

в) актор з ім'ям, індекс якого дорівнює номеру стовпця першого одиничного елемента в поточному рядку, додати в даний потік команд, а всі одиничні елементи даного стовпця обнулити;

г) якщо нове ім'я актора у потоці команд вже було додане до нього раніше, тоді закінчити формування чергового потоку відповідно до п. 3з;

д) в іншому випадку перейти до рядка, номер якого відповідає номеру стовпця, визначеного в п. 3в;

е) якщо рядок не містить одиничних елементів, закінчити формування чергового потоку команд відповідно до п. 3з;

ж) якщо рядок містить одиничні елементи, то продовжити формування чергового потоку відповідно до даного алгоритму, починаючи з п. 3б;

з) отриманому потоку присвоїти відповідний ідентифікатор СФК (якщо потік перший, то  $M_0$ , в інших випадках -  $M_{i+1}$ , де  $M_i$  – ідентифікатор попереднього потоку); перейти до наступного рядка для формування наступного потоку команд;

4. Якщо в результаті проходження матриці в ній залишились одиничні елементи, то прохід повторити з п. 3а; в іншому випадку виконувати п. 5.

5. Скласти перелік акторів  $I_i$  у відповідності з форматом (3), враховуючи зв'язки, задані матрицею суміжності, та ідентифікатори  $M_i$ .

6. Скласти перелік даних  $D_i$  у відповідності з форматом (4).

При виконанні алгоритму реальну кількість ОМ і СФК у системі не враховується. Внаслідок виконання алгоритму формується максимально можлива кількість паралельних потоків команд. Якщо в системі є лише одна СФК, то ідентифікатори  $M_i$  системою ігноруються. За наявності кількох СФК, кількість яких вибирається за формулою  $k = 2^j$  ( $j = 1, 2, 3, \dots$ ), реально буде формуватися стільки потоків команд, скільки СФК є в системі. При розподілі об'єктів (акторів і даних) між СФК ураховуються тільки  $\lfloor \log_2 k \rfloor$  молодших розрядів двійкових кодів ідентифікаторів  $M_j$ . Наприклад, для множини ідентифікаторів  $\{M_0, M_1, M_2, M_3, M_4, M_5\}$  відповідно з кодами  $\{00, 001, 010, 011, 100, 101\}$  за наявності в системі двох СФК ураховується тільки один молодший розряд кодів, а при чотирьох СФК – два молодші розряди. Відповідно в системі будуть формуватися два або чотири потоки команд.

## Висновки

Запропонований метод формування паралельних потоків команд у потокових системах має ряд переваг порівняно з реалізацією програмного керування паралельними обчисленнями.

Підготовка задачі зводиться до складання графа і розробки бібліотеки функцій. Керуючі слова й дані формуються автоматично компілятором. Немає необхідності попередньо враховувати кількість обчислювачів у системі, виділяти паралельні гілки і забезпечувати синхронізацію обміну даними між ними. Динамічний розподіл операцій дозволяє виявити безпосередньо в процесі обчислень прихований паралелізм, пов'язаний з різним часом обробки даних у різних гілках алгоритмів.

Обчислювальні модулі можуть виконувати операції, що належать до різних задач, у будь-якій послідовності. У зв'язку з цим у системі можуть одночасно розв'язуватися незалежні задачі, причому, немає необхідності синхронізації виконання задач.

Порівняно з відомим методом формування паралельних потоків команд запропонований метод розширює галузь застосування потокових систем за рахунок реалізації алгоритмів з циклами. Це підвищує ефективність обробки даних у потокових системах.

### Література

1. *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
2. *Dennis J. B., Missunas D. P.* A preliminary architecture for basic data flow processor// Proc. 2nd Annual Symp. Comput. Stockholm, May 1975. N. Y. IEEE. – 1975. – P. 126 – 132.
3. *Silva J.G.D., Wood J.V.* Design of processing subsystems for Manchester data flow computer // IEEE Proc. N.Y. – 1981. – Vol. 128, N 5. – P. 218 – 224.
4. *Watson R., Guard J.* A practical data flow computer // Computer. – 1982. – Vol. 15, N 2.– P. 51 – 57.
5. *Hogenauer E.B., Newbold R.F. Inn Y.T.* DDSF – a data flow computer for signal processing/ Proc. Int. Conf. Parall. Process. Ohio, August 1982. N.Y. // IEEE. – 1982. – P. 126 – 133.
6. *Жабина В.В.* Повышение эффективности параллельной обработки данных на уровне операций в потоковых системах // Вісник Національного технічного університету України “Київський політехнічний інститут”, Інформатика, управління та обчислювальна техніка, : Зб. наук. пр. – К.: „ВЕК+”. – 2008. – №49. – С. 112-116.