ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1994

Seria: INFORMATYKA z. 26 Nr kol. 1255

Stefan WĘGRZYN

PRZYSPIESZENIE REALIZACJI ALGORYTMÓW W SYSTEMACH STEROWANYCH PRZEPŁYWEM ARGUMENTÓW

Streszczenie. W pracy podano określenie przyspieszenia S realizacji algorytmów w systemach sterowanych przepływem argumentów i wskazano możliwości jego obliczenia wprost na podstawie macierzy zmiennych formy kanonicznej danego algorytmu. Może to być podstawą projektowania systemów sterowanych przepływem argumentów o bardzo krótkich czasach realizacji.

THE SPEEDUP OF ALGORITHMS EXECUTED IN SYSTEMS CONTROLLED BY FLOW OF ARGUMENTS

Summary. This paper introduces a new definition of the speedup S of realization of algorithm in argument-flow control systems. The possibilities of its calculation based directly on a matrix of variables of algorithm's canonical form are presented. It can be a base of argument-flow control systems with a very short realization time.

Pracę wykonano w ramach grantu KBN nr 3 P406 011 04

ACCÉLERATION DE LA RÉALISATION DES ALGORYTHMES DANS LES SYSTÈMES COMMANDÉS PAR LE FLUX DES ARGUMENTS

Resumé. L'article présente l'definition de la accélération de la réalisation des algorythmes dans les systèmes commandés par le flux des argumcnts et la possibilité de son calcule directement sur la base de la matrice de voriables de la forme cannonique. Ceci peut servire comme la base de la synthese des systèmes rapides commandés par le flux des arguments.

1. Wstęp

W przypadku równoległej realizacji algorytmów spotyka się pojęcie przyspieszenia S realizacji algorytmu. To przyspieszenie określa się jako stosunek czasu sekwencyjnej realizacji wszystkich występujących w algorytmie operacji do czasu realizacji tego algorytmu przy zastosowaniu trybu równoległego. Można np. spotkać tak zwane prawo Amdahla [1] w postaci:

$$S \leq \frac{1}{f+(1-f)/p}, \tag{1}$$

gdzie:

f – część operacji, która w danym algorytmie musi być wykonana w trybie sekwencyjnym, przy czym $0 \leq f \leq 1$,

p – liczba procesorów użytych w procesie zrównoleglania.

To tak zwane prawo Amdahla opiera się na nieuzasadnionym założeniu, że jeśli użyć p – procesorów dla realizacji w trybie równoległym części $(1-f)$ algorytmu, to czas jej realizacji będzie równy $(1-f)/p$, to znaczy, że dla $p = \infty$ czas realizacji części równoległej będzie równy zero, a czas realizacji części, która musi być zrealizowana w trybie sekwencyjnym, pozostanie równy f.

W niniejszej pracy podamy inne określenie przyspieszenia S realizacji algorytmu w trybie równoległym, a mianowicie:

$$S = \frac{l}{l_n}, \tag{2}$$

gdzie:

l – całkowita liczba operacji występujących w danym algorytmie,

l_n – liczba operacji uogólnionych, to jest takich, że operacje wykonywane równolegle liczy się jako jedną.

Wykażemy też, że w przypadku systemów sterowanych przepływem argumentów, wartość S można ocenić wprost na podstawie macierzy zmiennych formy kanonicznej danego algorytmu.

2. Forma kanoniczna algorytmu i jej macierz zmiennych

Algorytmem w postaci kanonicznej nazywamy [2] algorytm sprowadzony do uporządkowanej sekwencji instrukcji podstawienia.

Przez instrukcję podstawienia rozumiemy instrukcję O postaci: za zmienną y podstaw wynik pewnej operacji na argumentach, to jest danych a i zmiennych x, co zapisujemy:

$$y = O(a, x) \tag{3}$$

Pojęcie macierzy zmiennych formy kanonicznej zilustrujemy na przykładzie algorytmu o następującej postaci kanonicznej:

$$\begin{aligned} y_1 &= O_1(a_1) \\ y_2 &= O_2(y_1, a_2) \\ y_3 &= O_3(y_1, a_3) \\ y_4 &= O_4(y_1, y_2, y_3), \end{aligned} \tag{4}$$

gdzie:

a_1, a_2, a_3 – dane,

y_1, y_2, y_3, y_4 – zmienne.

Macierz zmiennych formy kanonicznej będzie miała w tym przypadku postać:

	y_1	y_2	y_3	y_4
y_1		×	×	×
y_2				×
y_3				×
y_4				

(5)

Z macierzy (5) odczytujemy, że w tym algorytmie y_1 staje się argumentem dla y_2, y_3, y_4, a y_2 i y_3 stają się argumentami dla y_4.

Innym przykładem form kanonicznych algorytmów może być algorytm typu embrionalnego (6).

$$\begin{aligned} y_1 &= O_1(a_1) \\ y_2 &= O_2(y_1, a_2) \\ y_3 &= O_3(y_1, a_3) \\ y_4 &= O_4(y_1, a_4) \end{aligned} \tag{6}$$

Jego macierz zmiennych ma postać (7):

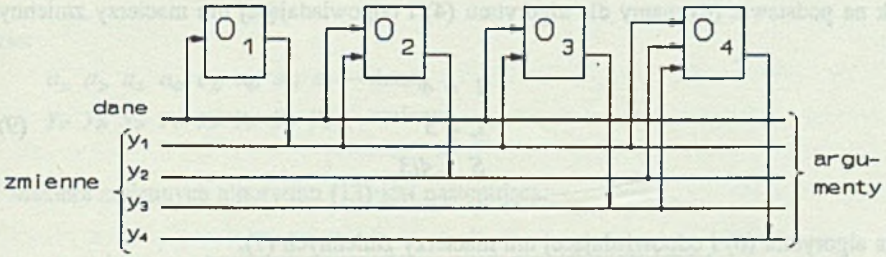
	y_1	y_2	y_3	y_4
y_1		×	×	×
y_2				
y_3				
y_4				

(7)

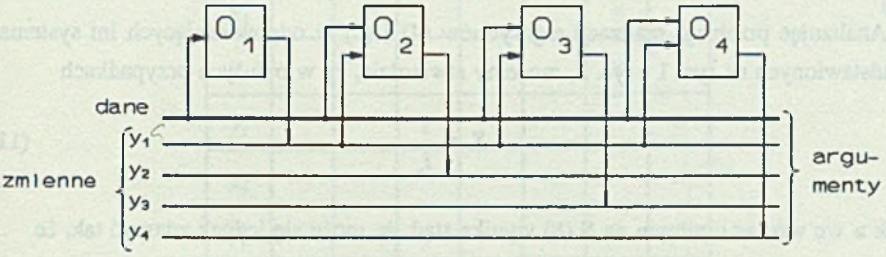
Z macierzy tej odczytujemy, że zmienna y_1 staje się argumentem dla zmiennych y_2, y_3, y_4.

3. Przyspieszenie realizacji algorytmów w sytemach sterowanych przepływem argumentów

Z postaci macierzy zmiennych np. (5) i (7) wynikają bezpośrednio struktury odpowiednich systemów sterowanych przepływem argumentów, które przedstawione są na rys. 1 i na rys. 2.



Rys. 1. System sterowany przepływem argumentów realizujący algorytm (5)
Fig. 1. A System controlled by flow of arguments and executing algorithm (5)



Rys. 2. System sterowany przepływem argumentów realizujący algorytm (6)
Fig. 2. A System controlled by flow of arguments and executing algorithm (6)

Zauważmy, porównując struktury rys. 1 i rys. 2 z macierzami (5) i (7), że **jeżeli kolumny macierzy zmiennych są identyczne, to operacje odpowiadające tym kolumnom wykonywane są równolegle.** Wynika stąd następująca postać wzoru na przyspieszenie S:

$$S \geq \frac{l}{l_r}, \tag{8}$$

gdzie:

l – liczba kolumn w macierzy zmiennych,

l_r – liczba różnych kolumn w macierzy zmiennych (kolumny identyczne liczy się jako jedna).

I tak na podstawie (8) mamy dla algorytmu (4) i odpowiadającej mu macierzy zmiennych (5):

$$\begin{aligned} l &= 4 \\ l_r &= 3 \\ S &\geq 4/3 \end{aligned} \tag{9}$$

a dla algorytmu (6) i odpowiadającej mu macierzy zmiennych (7):

$$\begin{aligned} l &= 4 \\ l_r &= 2 \\ S &\geq 2 \end{aligned} \tag{10}$$

Analizując przebiegi realizacji algorytmów (4) i (5) w odpowiadających im systemach przedstawionych na rys. 1 i rys. 2, możemy stwierdzić, że w obu tych przypadkach

$$S = \frac{l}{l_r} \tag{11}$$

Znak $\geq$ we wzorze ogólnym na S (8) wynika stąd, że może się jednak zdarzyć tak, że

$$S > \frac{l}{l_r} \tag{12}$$

Ma to miejsce wówczas, gdy operacje odpowiadające poszczególnym kolumnom będą w systemie sterowanym przepływem argumentów wykonywane równolegle, mimo tego że kolumny nie są identyczne. Wówczas, gdy kolumny nie są identyczne, ale poprzednia staje się argumentem dla następnej, operacje odpowiadające tym kolumnom będą w systemie sterowanym przepływem argumentów wykonywane równolegle i te dwie kolumny należy policzyć jako jedną. Omówimy to na przykładzie następującego algorytmu:

$$
\begin{aligned}
y_1 &= O_1(a_1, a_2) \\
y_2 &= O_2(a_3, a_4) \\
y_3 &= O_3(a_5, a_6) \\
y_4 &= O_4(a_7, a_8) \\
y_5 &= O_5(y_1, y_2) \\
y_6 &= O_6(y_3, y_4) \\
y_7 &= O_7(y_5, y_6),
\end{aligned} \tag{13}
$$

gdzie:

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$ – dane,

$y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8$ – zmienne.

Macierz zmiennych algorytmu (13) jest następująca:

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_1					×		
y_2					×		
y_3						×	
y_4						×	
y_5							×
y_6							×
y_7							

(14)

Kolumny y_5 i y_6 nie są identyczne, ale y_5 nie staje się argumentem dla y_6, operacje O_5 i O_6 będzie więc można zrealizować równolegle.

W tym przypadku jest więc liczba różnych kolumn w macierzy (14)

$$l_r = 4 \tag{15}$$

ale liczba operacji wykonywanych sekwencyjnie będzie równa:

$$l_r - 1$$

Stąd na podstawie (9)

$$S \geq \frac{l}{l_r} = \frac{7}{4}, \tag{16}$$

a uwzględniając to, że operacje właściwe dla kolumn y_5 i y_6 można wykonać równolegle, jest

$$S = \frac{l}{l_r - 1} = \frac{7}{3} \tag{17}$$

Warto zauważyć, że gdyby algorytm (13) przepisać w postaci (18):

$$\begin{aligned} y_1 &= O_1(a_1, a_2) \\ y_2 &= O_2(y_1, a_3) \\ y_3 &= O_3(y_2, a_4) \\ y_4 &= O_4(y_3, a_5) \\ y_5 &= O_5(y_4, a_6) \\ y_6 &= O_6(y_5, a_7) \\ y_7 &= O_7(y_6, a_8), \end{aligned} \tag{18}$$

to w macierzy zmiennych (19)

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_1		×					
y_2			×				
y_3				×			
y_4					×		
y_5						×	
y_6							×
y_7							

(19)

nie wystąpiłyby żadne kolumny identyczne, byłoby więc $l_r = l$ i w układzie nie otrzymalibyśmy żadnego przyspieszenia. Byłoby

$$S = \frac{l}{l_r} = 1 \tag{20}$$

4. Wnioski

Projektując dla realizacji danego algorytmu system sterowany przepływem argumentów i chcąc uzyskać w nim jak największe przyspieszenie S jego realizacji należy tak dobierać kanoniczną formę tego algorytmu, aby w jego macierzy zmiennych wystąpiła jak **najmniejsza liczba różnych kolumn.**

Przyspieszenie S zwiększają też takie przypadki, kiedy dwie sąsiednie kolumny są różne ale zmienna odpowiadająca poprzedniej nie staje się argumentem dla zmiennej odpowiadającej następnej.

LITERATURA

[1] Quinn M.J.: Designing Efficient Algorithms for Parallel Computers. McGrow-Hill, 1987.

[2] Węgrzyn S.: Systemy sterowane przepływem operacji i systemy sterowane przepływem argumentów. Zeszyty Naukowe Politechniki Śl., seria: Informatyka, z. 24, Gliwice 1993.

Recenzent: Doc. dr hab. inż. Adam Mrózek

Wpłynęło do Redakcji 27 stycznia 1994 r.

Abstract

This paper introduces a new definition of the speedup S of realization of algorithm in argument-flow control systems. It can be considered as a corrction of Amdahl's law which is used to estimatie the time of parallel execution of algorithms. The speedup defined here is the ratio l/l_n where l is the total number of operations in an algorithm and l_n is the number of generalised operations, i.e. the operations which parallel execution is possible and considered as the execution of one operation. The possibilities of its calculation based directly on a matrix of variables of algorithm's canonical form are presented. It can be a base of argument-flow control systems with a very short realization time.