

УДК 680.3

Г.А. ПОЛЯКОВ¹, Е.Г. ТОЛСТОЛУЖСКАЯ²¹Академия наук прикладной радиоэлектроники АНПРЭ, Украина, Россия²Харьковский национальный университет им. В.Н. Каразина, Украина

МЕТОД СИНТЕЗА ВРЕМЕННЫХ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ КЛАССА MPP

Приведено описание метода синтеза временных параллельных программ для вычислительных систем с распределенной памятью (MPP). Используемые исходные данные: Си – программа задачи; архитектура ЭВМ – однородная, топология коммутационной среды – n -мерный гиперкуб, количество процессоров NM , известно число портов приема/передачи данных процессора; задана конструкция (состав полей) сообщений; синхронизация – с помощью библиотечных средств; заданы значения длительностей выполнения различных операций; метод параллельной обработки – совмещение независимых операций. Отмечено, что разработанный метод позволяет автоматизировать синтез параллельных программ для ЭВМ класса MPP, содержащих в явном виде время начала выполнения множеств независимых операторов.

Ключевые слова: временная параллельная программа, вычислительные системы с распределенной памятью (MPP), методы параллельной обработки данных.

Введение

Ведущими специалистами в области параллельного программирования отмечается [1–3], что решающим значением для успеха использования существующих и будущих супер-ЭВМ (в первую очередь, в интересах критических систем реального времени) является повышение их надежности и эффективности на основе автоматизации синтеза параллельных программ.

Анализ показывает, что программист должен выполнять вручную наиболее сложные, творческие этапы параллельного программирования, которые определяют оптимальность параллельных программ, предельно допустимую сложность эффективно программируемых задач и сроки разработки параллельных программ [2, 3].

В статье представлен метод, позволяющий автоматизировать процесс синтеза временных параллельных программ для вычислительных систем класса MPP.

1. Постановка задачи

Исходная информация:

- Си – программа решаемой задачи;
- класс параллельной ЭВМ – однородная многопроцессорная ЭВМ с распределенной памятью (MPP);
- число процессоров NM – произвольное конечное ($NM \geq 2$);

- топология коммутационной среды – n -мерный единичный гиперкуб;

- синхронизация – с помощью библиотечных средств (директив, барьеров и т.п.);

- метод параллельной обработки – совмещение независимых операций;

- количество портов приема/передачи данных произвольного процессора $k_port \geq 1$;

- конструкция (состав полей) сообщений считается заданной, время обмена одним сообщением между смежными (соседними по топологии) процессорами (например, с помощью пары операторов «send – receive») t_c^0 ;

- значения t^0 (тип) длительностей выполнения различных типов «тип» арифметических, логических операций и операций обращения к распределенной памяти процессоров MPP ЭВМ (в тактах).

Требуется:

Синтезировать временную параллельную программу для ЭВМ класса MPP, логически эквивалентную исходной последовательной Си – программе и обеспечивающую выполнение одного из следующих требований:

- достижение потенциально возможного для задачи времени параллельного решения;

- достижение заданного времени решения задачи с оптимизацией равномерности загрузки используемых процессоров;

- достижение минимального времени решения задачи с учетом ограничений на количество используемых процессоров.

2. Этапы решения задачи

Определим параллельную временную программу, как конструкцию, в которой специфицированы в явном виде следующие категории информации [4]:

- спецификацию разделения задачи на подзадачи (фрагменты);
- спецификацию распределения фрагментов задачи на процессоры для выполнения;
- спецификацию распределения данных фрагментов по процессорам;
- спецификацию моментов начала параллельного выполнения независимых операторов каждого фрагмента;

- средства организации обмена данными между фрагментами (процессорами);
- средства синхронизации процессов (параллельных или последовательных), реализуемых соответствующим подмножеством процессоров MPP ЭВМ.

Обобщенный алгоритм синтеза параллельных MPP-программ представлен на рис. 1.

Проиллюстрируем результаты основных групп этапов алгоритма с помощью задачи (Си-программа представлена на рис. 2) для случая совмещения независимых операций при ограничении количества процессоров. ($NM = 2$).

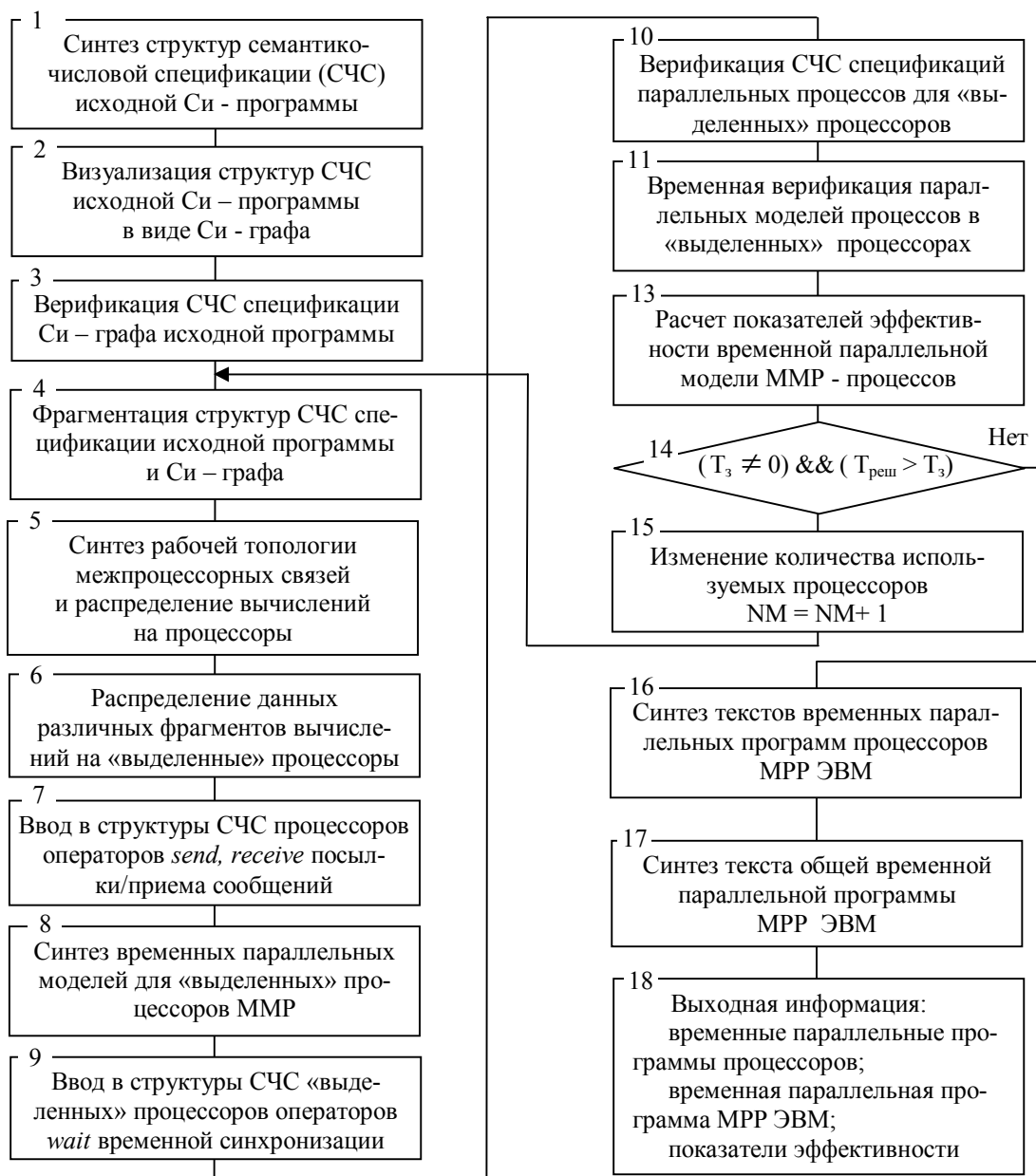


Рис. 1. Обобщенный алгоритм синтеза параллельных MPP-программ

Результаты проверки корректности выполнения этапов 1...3 представляет рис. 3. Структуры СЧС и Си – граф [4] вынесены в презентацию доклада. Результаты синтеза текстов временных программ «выделенных» процессоров (этапы 4...9) показаны на рис. 4,5. Параметры операторов send (1,k,2) / receive(1,k,2) : «адрес источника», «значения», «адрес приемника». Размещение данных в распределенной памяти процессоров MPP ЭВМ представлено в табл. 1.

```
include <stdio.h>
void main(void )
{ int a,b,c,r,k,l,m,p, s,t;
  scanf("%d %d %d\n",&a,&b,&c);
  k = a * b;
  l = b % a;
  if(k < a-c) { m = (k % 2) * 2;
               r = l * 2;
               p = k + 1;}
  else { p = 2 * l; r = l - k;
        m = p + l; }
  s = p - r;
  t = (m * 2) / a;
  printf("%4d %4d\n",s,t);
}
```

Рис. 2. Си – программа задачи

```
ТЕСТ КОРРЕКТНОСТИ ФАЙЛОВ:
максимальное количество элементов: 0 - 51
максимальное количество связей: 0 - 78
ТЕСТ СООТВЕТСТВИЯ ЧИСЛА СОПРЯЖЕННЫХ И ВНЕШНИХ СВЯЗЕЙ: ОК
ТЕСТ ЧИСЛА СВЯЗЕЙ ПО СОПРЯЖЕННЫМ ЭЛЕМЕНТАМ: ОК
ТЕСТ ЧИСЛА СВЯЗЕЙ ПО ВНЕШНИМ ЭЛЕМЕНТАМ: ОК
ТЕСТ СООТВЕТСТВИЯ ВЫВОДОВ ПО СОПРЯЖЕННЫМ ЭЛЕМЕНТАМ: ОК
ТЕСТ СООТВЕТСТВИЯ ВЫВОДОВ ПО ВНЕШНИМ ЭЛЕМЕНТАМ: ОК
```

Рис. 3. Результаты верификации СЧС спецификации и Си – графа исходной программы (этапы 1...3)

Таблица 1

Размещение данных в распределенной памяти процессоров MPP ЭВМ (этап 6)

ЗУ 1-го процессора		ЗУ 2-го процессора	
Адреса данных	Имена данных	Адреса данных	Имена данных
0	b		a
1	a	1	a1
2	c	2	k
3	k	3	l
4	a1	4	C2_
5	l	5	k1
6	C2_	6	m
7	r	7	p
8	p	8	m1
9	s	9	t

```
#include <stdio.h>
void main(void)
{
    int a,b,c,r;
    int k,l,m,p;
    int s,t;
    int a1,k1,m1;
    scanf("%d",&b);
    scanf("%d",&a);
    scanf("%d",&c);
    k = a * b;
    send(1,k,2);
    a1 = a - c;
    send(1,a1,2);
    l = b % a;
    send(1,l,2);
    if (k < a1)
    { r = l * 2; p = k + 1;
      wait (37.00);
    }
    else { p = 2 * l;
          send(1,p,2);
          r = l - k;
          wait (14.00);
        }
    s = p - r;
    printf(" %3d ",s);
    wait (46.00);
}
```

Рис. 4. Текст программы первого процессора MPP ЭВМ (этапы 4...9)

```
#include <stdio.h>
void main(void)
{
    int a,b,c,r;
    int k,l,m,p;
    int s,t;
    int a1,k1,m1;
    scanf("%d",&a);
    wait ( 7.00);
    a1 = receive(1,a1,2);
    k = receive(1,k,2);
    wait (21.00);
    l = receive(1,l,2);
    if (k < a1)
    { k1 = k % 2;
      m = k1 * 2;}
    else
    { wait (16.00);
      p = receive(1,p,2);
      m = p + l; }
    m1 = m * 2;
    t = m1 / a;
    printf(" %3d ",t);
}
```

Рис. 5. Текст программы второго процессора MPP ЭВМ (этапы 4...9)

Временные программы процессоров MPP выполнения операторов (этап 16) представлены ЭВМ с явным заданием моментов NT начала в табл. 2.

Таблица 2

Моменты NT начала выполнения операторов процессоров MPP ЭВМ

NN	Команды 1-го процессора			NT	NN	Команды 2-го процессора			NT
	oper	A1	A2			oper	A1	A2	
0	scanf	0	-1	1.00	0	scanf	0	-1	1.00
1	scanf	1	-1	1.00	1	wait(7.00);			3.00
2	scanf	2	-1	1.00	2	receive(1,1,2)			10.00
3	*	1	0	3.00	3	=	1	-1	19.00
4	=	3	-1	13.00	4	receive(1,2,2)			19.00
5	send(1,3,2)			15.00	5	=	2	-1	28.00
6	-	1	2	3.00	6	wait(21.00);			30.00
7	=	4	-1	4.00	7	receive(1,3,2)			51.00
8	send(1,4,2)			6.00	8	=	3	-1	60.00
9	%	0	1	10.00	9	<	2	1	62.00
10	=	5	-1	45.00	10	upl	11	16	63.00
11	send(1,5,2)			47.00	11	%	2	4	64.00
12	<	3	4	62.00	12	=	5	-1	99.00
13	upl	14	20	63.00	13	*	5	4	101.00
14	*	5	6	64.00	14	=	6	-1	111.00
15	=	7	-1	74.00	15	bp	22	-1	113.00
16	+	3	5	64.00	16	wait(16.00);			64.00
17	=	8	-1	65.00	17	receive(1,7,2)			80.00
18	wait(37.00);			76.00	18	=	7	-1	89.00
19	bp	27	-1	113.00	19	+	7	3	91.00
20	*	6	5	64.00	20	=	6	-1	92.00
21	=	8	-1	74.00	21	bp	22	-1	94.00
22	send(1,8,2)			76.00	22	*	6	4	114.00
23	-	5	3	64.00	23	=	8	-1	124.00
24	=	7	-1	65.00	24	/	8	0	126.00
25	wait(14.00);			80.00	25	=	9	-1	161.00
26	bp	27	-1	94.00	26	print	9	-1	163.00
27	-	8	7	114.00	27	bp	28	-1	164.00
28	=	9	-1	115.00	28	stop	-1	-1	165.00
29	print	9	-1	117.00					
30	wait(46.00);			118.00					
31	bp	32	-1	164.00					
32	stop	-1	-1	165.00					

Заключение

1. В настоящее время проблема разработки эффективного параллельного программного обеспечения для известных и будущих суперЭВМ стала цен-

тральной проблемой параллельной компьютерной техники.

2. Разработанный метод обеспечивает формализацию синтеза временных параллельных MPP-программ, исходя из традиционных последо-

вательних Си-программ и известной конфигурации ЭВМ, удовлетворяющих предъявленным требованиям/ограничениям (на время решения, доступный ресурс и т.д.). Традиционные последовательные и параллельные программы являются частными случаями временных параллельных программ.

3. Изложенный метод представляет собой основу создания инструментальных средств автоматического синтеза статических и временных параллельных программ для известных параллельных вычислительных систем класса МРР и перспективных Адаптивных Самоорганизующихся Вычислительных Систем.

Литература

1. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002. – 608 с.

2. Немнюгин С.А. Параллельное программирование для многопроцессорных вычислительных систем. / С.А. Немнюгин, О.Л. Стесик. – СПб.: БХВ-Петербург, 2002. – 400 с.

3. Корнеев В.В. Параллельные вычислительные системы / В.В. Корнеев. – М.: Нолидж, 1999. – 320 с.

4. Поляков Г.А. Технология проектирования времяпараметризованных мультипараллельных программ как стратегия развития систем параллельного проектирования. / Г.А. Поляков, Е.Г. Толстолужская. // *Радиоелектронні і комп'ютерні системи* – Х.:ХАІ, 2009. – Вип. 6(40). – С. 166-171.

5. Поляков Г.А. Метод формального архитектурно-ориентированного проектирования временных параллельных программ для ЭВМ с симметричной мультипроцессорной обработкой данных / Г.А. Поляков, Е.Г. Толстолужская. // *Збірник наукових праць Харківського університету Повітряних Сил*. – Х.: ХУПС, 2008. – Вип. 3(18) – С. 118-121.

Поступила в редакцию 1.02.2010

Рецензент: д-р техн. наук, проф., ведущий научный сотрудник Е.Л. Казаков, Харьковский университет Воздушных Сил, Харьков, Украина.

МЕТОД СИНТЕЗУ ЧАСОВИХ ПАРАЛЕЛЬНИХ ПРОГРАМ ДЛЯ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ КЛАСУ МРР

Г.О. Поляков, О.Г. Толстолужка

Наведено опис метод синтезу часових паралельних програм для обчислювальних систем з розподіленою пам'яттю (МРР). Використовувані вихідні дані: Си-програма задачі; архітектура ЕОМ – однорідна, топологія комутаційного середовища – n-вимірний гіперкуб, кількість процесорів NM, відома кількість портів прийому/передачі даних процесора; задана конструкція (склад полів) повідомлень; синхронізація – за допомогою бібліотечних засобів; задані значення часу виконання різних операцій; метод паралельної обробки – сполучення незалежних операцій. Відзначено, що розроблений метод дозволяє автоматизувати синтез паралельних програм для ЕОМ класу МРР, що містять у явному виді час початку виконання множин незалежних операторів.

Ключові слова: часова паралельна програма, обчислювальні системи з розподіленою пам'яттю, методи паралельної обробки даних.

METHOD OF SYNTHESIS OF TEMPORAL PARALLEL PROGRAMS FOR COMPUTER SYSTEMS OF MPP CLASS

G.A. Polyakov, H.G. Tolstolughskaya

The method of synthesis of the temporal parallel programs is resulted for the computer system with the distributed memory (MPP) is resulted. Used basic data: A C is the program of task; COMPUTER architecture – homogeneous, a topology of commutation environment is the n-measured hypercube, amount of processors of NM, the number of ports of adopting/transmission information of processor is known; the construction (composition of the fields) of reports is set; synchronization – by facilities of libraries; the values of durations of implementation of different operations are set; a method of the simultaneous processing is combination of independent operations. It is marked, that the developed method allows to automatize the synthesis of the parallel programs for COMPUTERS of class of MRR, containing in an obvious kind time began implementations of great numbers of independent operators.

Keywords: temporal parallel program, computer systems with the distributed memory, methods of the simultaneous processing of data.

Поляков Геннадий Алексеевич – д-р техн. наук, проф., академик, Академия наук прикладной радиоэлектроники, Москва, Россия.

Толстолужская Елена Геннадиевна – канд. техн. наук, ст. научн. сотр., доц. кафедры теоретической и прикладной системотехники факультета компьютерных наук, Харьковский национальный университет им. В.Н. Каразина, Харьков, Украина, e-mail: tda_ua@pochtamtu.ru.