UDC 681.31

H.G. Tostoluzhskaya

V.N. Karazin Kharkiv National University, Ukraine

SYNTHESIS OF TIME PARAMETERIZED PARALLEL PROGRAMS FOR COMPUTER SYSTEMS OF CLASS OF MPP

The method of synthesis of the time parameterized (timing) parallel programs for the computer system with the distributed memory (MPP) is considered. Used basic data: C-program of task; computer architecture - a homogeneous topology of commutation environment is the n-measured hypercube, number of processors of NM, the structure of sending and receiving commands; synchronization - by facilities of libraries; the values of durations of implementation of different operations are set; the method of the simultaneous processing is the 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 MPP, which contain in an obvious kind time of begging of the implementations of great numbers of independent operators.

Keywords: *time parameterized parallel program, computer systems with the distributed memory (MPP), methods of the simultaneous processing of data.*

Introduction

Leading experts in the field of parallel programming note [1 – 3], that critical key for the success of using existing and future supercomputers (primarily in the interest of critical real-time systems) is to increase the reliability and efficiency through automation of synthesis of parallel programs. The analysis shows that the programmer must do manually the most complex and creative phases of parallel programming, which determines the optimal parallel programs, the maximum allowable complexity effectively programmable tasks and timeframe for developing parallel programs [2, 3]. The article presents a method allowing to automate the process of synthesis temporal parallel programs for computer systems of class of MPP.

1. Formulation of the problem

1.1. Initial information:

• The values of performance in parallel - pipeline device (while solving the problem, the value of the interval clock / clock frequency, the number of functional units of different types / the total gate complexity / the cost) To synthesize a temporal parallel computer program class MPP, logically equivalent to the original serial C - program and provide execution one of the following requirements:

• C-program does the task;

• the class of parallel computer – homogeneous multiprocessor computers with distributed memory (MPP);

• the number of processors NM – finite (NM - 2);

• the topology of commutation environment is the n-measured hypercube;

• the synchronization – by facilities of libraries (directives, barriers and others);

• the method of the simultaneous processing is the combination of independent operations;

• the construction (the composition of fields) messages is given, the exchange time of a message between;

• adjacent (neighboring on the topology) processors (for example, using a pair of operators «send – receive»);

• the value t^0 (typ) duration of the execution of various types of «typ» arithmetic, logical operations and operations of reversion to distributed memory processors MPP computers (in clock cycles).

1.2. Required:

• The values of performance in parallel – pipeline device (while solving the problem, the value of the interval clock / clock frequency, the number of functional units of different types / the total gate complexity / the cost) To synthesize a temporal parallel computer program class MPP, logically equivalent to the original serial C - program and provide execution one of the following requirements:

• the achievement of the potential opportunities for the task time of parallel solution;

• the achievement of given time of the solution task with optimization even loading of using processors;

• the achievement of minimum time solving the task with considering the limitations on the number of using processors.

2. Steps of solving the problem

Define the parallel temporal program like a construction in which the following categories of information explicitly specified [4 – 6]:

• the specification of the division of tasks into subtasks (fragments);

• the specification of fragments distribution of task to processors for execution;

• the specification of the distribution of these fragments to processors;

• the specification of the moments of the beginning of the parallel execution of independent operators of each fragment;

• the resources of exchange data between fragments (processors);

• the synchronization of tools of process (parallel or serial) implemented appropriate subset processors MPP computers.

The generic synthesis algorithm of parallel MPP programs is presented in Fig. 1.

We illustrate the results of the main groups of the stages of the algorithm with the task (C-program is shown in Fig. 2) for a combination of independent operations by limiting the number of processors (NM = 2). SNS structures and C - graph are delivered in the Tabl. 1, Tabl. 2 and Fig. 3.

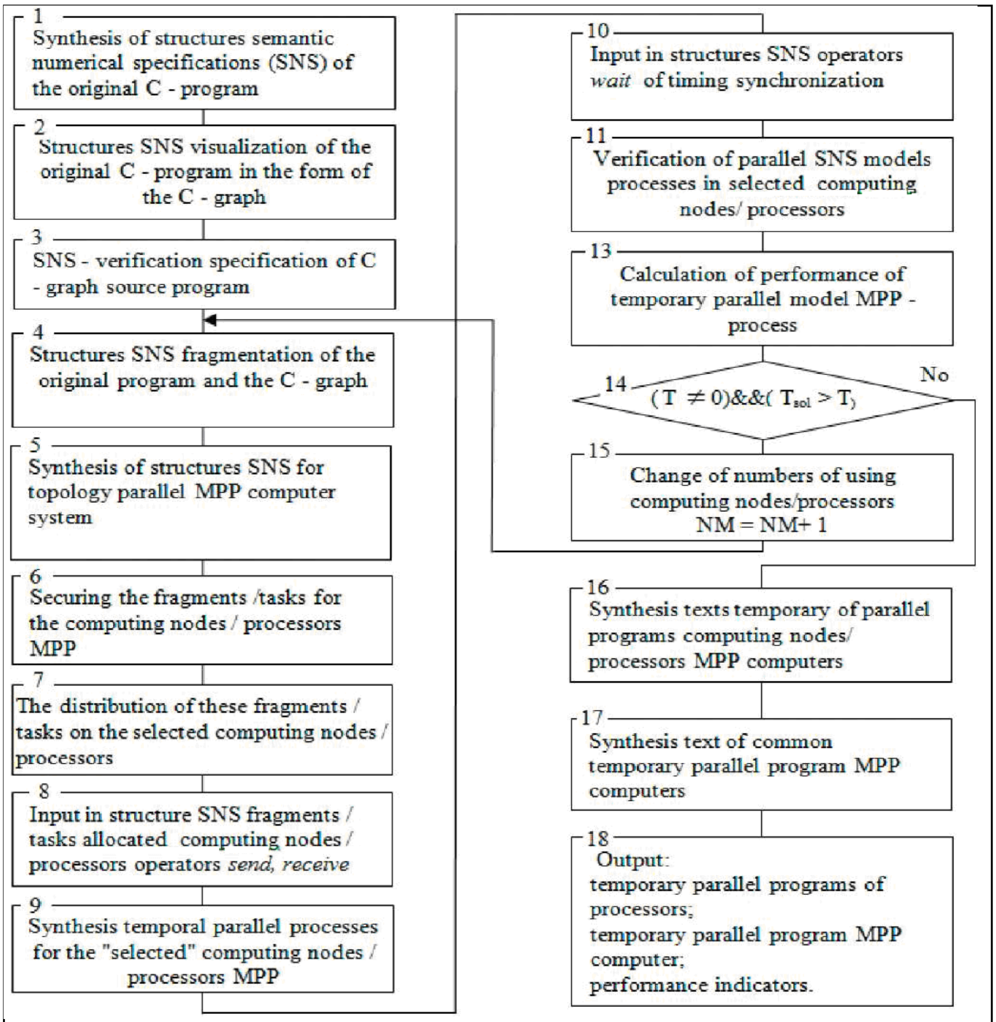

Fig. 1. The generic algorithm synthesis of parallel models and MPP programs

```
#include <stdio.h>
void main(void )
{
int a,b,c,r,k,l,m,p,s,t;

    scanf("%d",&a);
    scanf("%d",&b);
    scanf("%d",&c);
    k = a * b;
l = b % a;
if (k < (a - c))
{
    m = (k % 2) * 2;
    r = l * 2;
    p = k + l;
else
{
    p = 2 * l;
        r = l - k;
        m = p + l;
    }
    s = p - r;
    printf(" %3d ",s);
    t = (m * 2) / a;
    printf(" %3d ",t);
}
```

Fig. 2. C-program of task

Table 1

BF structure of semantic - numeric specification of operators in C – programs

N	MET	TYP	NSJ	SJD	BJ	NWJ	WJD	MP1	MP2	VH	VIH	RES
0	0	58	-1	0	0	0	1	0	0	0	1	a_in
1	0	58	-1	0	0	1	1	0	0	0	1	b_in
2	0	58	-1	0	0	2	1	0	0	0	1	c_in
3	0	47	-1	0	0	3	1	0	0	0	2	a
4	0	47	-1	0	0	4	1	0	0	0	2	b
5	0	47	-1	0	0	5	1	0	0	0	2	c
6	0	47	-1	0	0	6	2	0	0	0	2	r
7	0	47	-1	0	0	8	1	0	0	0	2	k
8	0	47	-1	0	0	9	1	0	0	0	2	1
9	0	47	-1	0	0	10	2	0	0	0	2	m
10	0	47	-1	0	0	12	2	0	0	0	2	P
11	0	47	-1	0	0	14	1	0	0	0	2	s
12	0	47	-1	0	0	15	1	0	0	0	2	t
13	0	12	0	2	0	16	4	0	0	2	1	=
14	0	12	2	2	0	20	2	0	0	2	1	=
15	0	12	4	2	0	22	1	0	0	2	1	=
16	0	3	6	2	0	23	1	0	0	2	1	*
17	0	12	8	2	0	24	4	0	0	2	1	=
18	0	5	10	2	0	28	1	0	0	2	1	%
19	0	12	12	2	0	29	5	0	0	2	1	=
20	0	2	14	2	0	34	1	0	0	2	1	
21	0	25	16	2	0	35	1	0	0	2	1	<
22	0	51	18	1	0	36	5	1	2	1	2	upl
23	0	57	-1	0	1	41	5	0	0	0	1	C2_
24	1	5	19	3	1	46	1	0	0	3	1	%
25	0	3	22	2	1	47	1	0	0	2	1	*
26	0	12	24	2	1	48	2	0	0	2	2	=
27	0	3	26	3	1	50	1	0	0	3	1	*
28	0	12	29	2	1	51	2	0	0	2	2	=
29	0	1	31	3	1	53	1	0	0	3	1	+
30	0	12	34	2	1	54	2	0	0	2	2	=
31	0	50	36	3	1	56	1	3	0	3	1	bp
32	2	3	39	3	2	57	1	0	0	3	1	*
33	0	12	42	2	2	58	2	0	0	2	1	=
34	0	2	44	3	2	60	1	0	0	3	1	-
35	0	12	47	2	2	61	2	0	0	2	2	=
36	0	1	49	2	2	63	1	0	0	2	1	+
37	0	12	51	2	2	64	2	0	0	2	2	=
38	0	50	53	2	2	66	1	3	0	2	1	bp
39	3	54	55	2	3	67	1	0	0	2	1	1.0
40	0	53	57	2	3	68	1	0	0	2	1	a. o
41	0	53	59	2	3	69	1	0	0	2	1	a. o
42	0	53	61	2	3	70	1	0	0	2	1	a. o
43	0	2	63	3	3	71	1	0	0	3	1	-
44	0	12	66	2	3	72	2	0	0	2	2	=
45	0	3	68	2	3	74	1	0	0	2	1	*
46	0	4	70	2	3	75	1	0	0	2	1	/
47	0	12	72	2	3	76	2	0	0	2	2	
48	0	50	74	2	3	78	1	980	0	2	1	bp
49	980	49	76	1	4	-1	0	0	0	1	0	1 stop
50	0	48	77	1	4	-1	0	0	0	1	0	s_out
51	0	48	78	1	4	-1	0	0	0	1	0	t_out

In the structures of SNS BF, BFM (Tabl. 1, Tabl. 3) N – number j « operator » Pj C programs (names starting / output data of the login data input and output, user IDs); MET – statement labels ; TYP – types of operators ; NSJ (Pj) – pointers to the beginning of a chain of operators Pi, is the operand to Pj (conjugate set); SJD (Pj) – the number of sjd (Pj) «attended» for operators Pj Pi; BJ – set of parts of the C-program (fragments operators); NWJ (Pj) – nwj pointers to the beginning of the chain of «external» operators using the result of the operator Pj; WJD (Pj) – the number of wjd «external» to the operators Pj Pi; MP1, MP2 – label operators; VH and VIH – the number of inputs (operands) and the number of outputs (output) of the operators; RES – name data and instructions / functions; BJF – number of steps (fragments) in parallel-pipeline model; NT – moments of the beginning operations; NY – temporary accommodation tiers of the model (Tabl. 4, Fig. 7).

Table 2

CF structure of semantic – numeric specification of C – program communication operators

N	JSD	SPJD	SNWIH	SNWHO	TSS	JWD	WPJD	WNWHO	WNWIH	TVS	N	JSD	SPJD	SNWIH	SNWHO	TSS	JWD	WPJD	WNWHO	WNWIH	TVS
0	1	0	0	0	0	-1	13	0	0	0	39	40	23	0	0	0	40	32	2	1	1
1	-1	3	1	1	2	-1	14	0	0	0	40	41	19	0	1	0	-1	34	2	1	1
2	3	1	0	0	0	-1	15	0	0	0	41	-1	22	1	2	1	42	24	1	0	0
3	-1	4	1	1	2	-1	13	1	1	2	42	43	10	1	1	2	43	25	1	0	0
4	5	2	0	0	0	-1	14	1	1	2	43	-1	32	0	0	0	44	27	1	0	0
5	-1	5	1	1	2	-1	15	1	1	2	44	45	19	0	0	0	45	32	0	0	0
6	7	13	0	0	0	7	28	1	1	2	45	46	17	0	1	0	-1	45	1	0	0
7	-1	14	0	1	0	-1	35	1	1	2	46	-1	22	1	2	1	-1	25	0	0	0
8	9	7	1	1	2	-1	17	1	1	2	47	48	6	1	1	2	-1	26	0	0	0
9	-1	16	0	0	0	-1	19	1	1	2	48	-1	34	0	0	0	49	31	2	1	1
10	11	14	0	0	0	11	26	1	1	2	49	50	33	0	0	0	-1	40	0	0	0
11	-1	13	0	1	0	-1	37	1	1	2	50	-1	19	0	1	0	-1	28	0	0	0
12	13	8	1	1	2	13	30	1	1	2	51	52	9	1	1	2	52	31	1	1	1
13	-1	18	0	0	0	-1	33	1	1	2	52	-1	36	0	0	0	-1	41	0	0	0
14	15	13	0	0	0	-1	44	1	1	2	53	54	37	1	0	1	-1	30	0	0	0
15	-1	15	0	1	0	-1	47	1	1	2	54	-1	35	1	1	1	55	31	0	1	1
16	17	17	0	0	0	17	16	0	0	0	55	56	38	0	1	1	-1	42	0	0	0
17	-1	20	0	1	0	18	18	1	0	0	56	-1	31	0	0	1	-1	39	0	0	1
18	-1	21	0	0	0	19	20	0	0	0	57	58	37	0	1	0	-1	33	0	0	0
19	20	17	0	0	0	-1	46	1	0	0	58	-1	26	0	0	0	59	36	0	0	0
20	21	23	0	1	0	21	16	1	0	0	59	60	35	0	1	0	-1	42	1	0	0
21	-1	22	0	2	1	-1	18	0	0	0	60	-1	28	0	0	0	-1	35	0	0	0
22	23	24	0	0	0	-1	20	1	0	0	61	62	33	0	1	0	62	38	1	1	1
23	-1	23	0	1	0	-1	17	0	0	0	62	-1	30	0	0	0	-1	41	1	0	0
24	25	9	1	1	2	25	21	0	0	0	63	64	42	0	0	0	-1	37	0	0	0
25	-1	25	0	0	0	26	24	0	0	0	64	65	41	0	1	0	65	38	0	1	1
26	27	19	0	0	0	27	29	0	0	0	65	-1	39	0	2	1	-1	40	1	0	0
27	28	23	0	1	0	-1	34	1	0	0	66	67	11	1	1	2	-1	39	1	0	1
28	-1	22	0	2	1	-1	19	0	0	0	67	-1	43	0	0	0	-1	43	2	0	1
29	30	6	1	1	2	30	27	0	0	0	68	69	40	0	0	0	-1	45	0	0	0
30	-1	27	0	0	0	31	29	1	0	0	69	-1	23	0	1	0	-1	43	1	0	0
31	32	17	0	0	0	32	32	1	0	0	70	71	45	0	0	0	-1	43	0	0	0
32	33	19	0	1	0	33	34	0	0	0	71	-1	13	0	1	0	-1	44	0	0	0
33	-1	22	0	2	1	-1	36	1	0	0	72	73	12	1	1	2	73	48	1	1	1
34	35	10	1	1	2	-1	21	1	0	0	73	-1	46	0	0	0	-1	50	0	0	0
35	-1	29	0	0	0	-1	22	0	0	0	74	75	47	1	0	1	-1	46	0	0	0
36	37	30	1	0	1	37	24	2	0	1	75	-1	44	1	1	1	-1	47	0	0	0
37	38	28	1	1	1	38	27	2	0	1	76	-1	48	0	0	1	77	48	0	1	1
38	-1	26	1	2	1	39	29	2	0	1	77	-1	44	0	0	0	-1	51	0	0	0

In the structures of SNS CF (Tabl. 2) NN – the line number in the structure of the CF bonds operators; SPJD (Pj) – numbers of operators Pi, which are the operands for Pj (attended set); JSD – a pointer to a continuation of the chain of operators set SPJD (j); SNWIH (Pi) – the output number of the operator Pi, attended to Pj; SNWHO (Pj) – input number of «external» to the operator Pi Pj; WPJD (Pj) – «external» set of operators Pi, using the result of the operator Pj; JWD – pointer continuation of the chain operators Pi «external» to Pj; WNWHO (Pi) – the input number of the operator Pi, «external» to Pj; WNWIH (Pj) – the output number of the operator Pj; TSS, TVS – «types» (according to, for management and for acquaintance) attended and external relations statements in C – programs / vertex C – Column / parallel model / digital apparatus (according to, for management of news).

The above is the result of performing steps 1 – 3 of the generalized algorithm (Fig. 1). The results of the synthesis steps 4 – 7 time parameterized fragmented parallel model C program objectives are SCHS VFM structure (Tabl. 3), Fig. 4 and Tabl. 4.

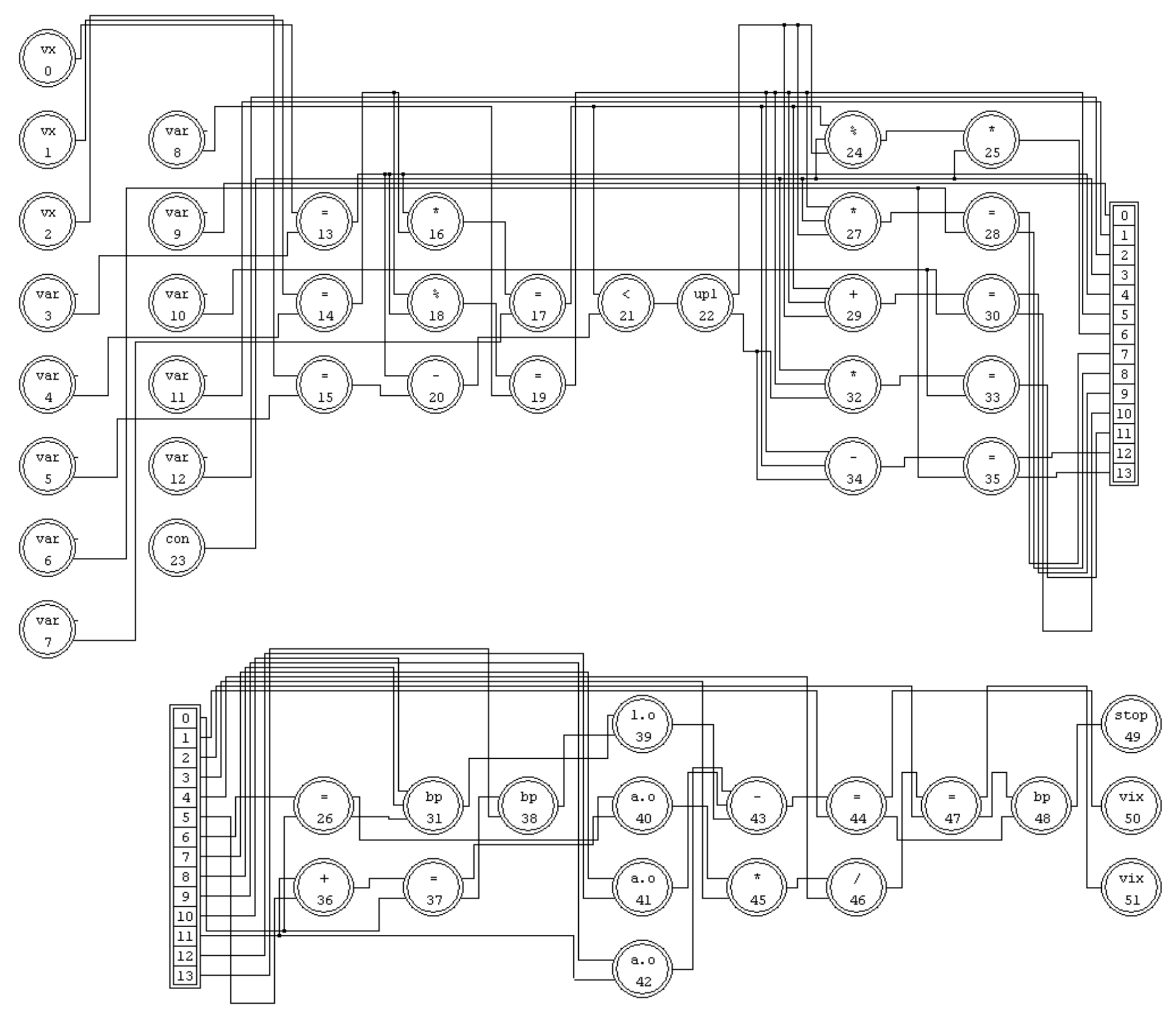

Fig. 3. C – graph of initial C - program

Table 3

Structure BFM of semantic – numeric specification of fragmentation timing model C – program operators

N	TYP	NSJ	SJD	BJ	NWJ	WJD	FRAG	SLO	NT	NY	RES	N	TYP	NSJ	SJD	BJ	NWJ	WJD	FRAG	SLO	NT	NY	RES
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	58	-1	0	0	0	1	1	0	0.00	0	a_in	29	1	32	3	1	53	1	1	0	6.00	6	+
1	58	-1	0	0	1	1	1	0	0.00	0	b_in	30	12	35	2	1	54	2	1	0	7.00	7	=
2	58	-1	0	0	2	1	1	0	0.00	0	c_in	31	50	37	4	1	56	1	1	0	10.00	10	bp
3	47	-1	0	0	3	1	1	0	0.00	0	a	32	3	41	3	2	57	1	1	0	6.00	6	*
4	47	-1	0	0	4	1	1	0	0.00	0	b	33	12	44	2	2	58	2	1	1	7.00	7	=
5	47	-1	0	0	5	1	1	0	0.00	0	c	34	2	46	3	2	60	1	1	0	6.00	6	--
6	47	-1	0	0	6	2	1	0	0.00	0	r	35	12	49	2	2	61	2	1	0	7.00	7	=
7	47	-1	0	0	8	1	1	0	0.00	0	k	36	1	51	2	2	63	1	2	0	8.00	8	+
8	47	-1	0	0	9	1	1	0	0.00	0	1	37	12	53	2	2	64	2	2	1	9.00	9	=
9	47	-1	0	0	10	2	2	0	0.00	0	m	38	50	55	2	2	66	1	1	0	10.00	10	bp
10	47	-1	0	0	12	2	1	0	0.00	0	P	39	54	57	2	3	67	1	1	0	11.00	11	1.0
11	47	-1	0	0	14	1	1	0	0.00	0	s	40	53	59	2	3	68	1	2	0	11.00	11	a.o
12	47	-1	0	0	15	1	2	0	0.00	0	t	41	53	61	2	3	69	1	1	0	11.00	11	a.o
13	12	0	2	0	16	4	1	0	1.00	1	=	42	53	63	2	3	70	1	1	0	11.00	11	a.o
14	12	2	2	0	20	2	1	0	1.00	1	=	43	2	65	3	3	71	1	1	0	12.00	12	-

Table 3 (end)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
15	12	4	2	0	22	1	1	0	1.00	1	=	44	12	68	2	3	72	2	1	0	13.00	13	=
16	3	6	2	0	23	1	1	0	2.00	2	*	45	3	70	2	3	74	1	2	0	12.00	12	*
17	12	8	2	0	24	4	1	1	3.00	3	=	46	4	72	2	3	75	1	2	0	14.00	14	/
18	5	10	2	0	28	1	1	0	2.00	2	%	47	12	74	2	3	76	2	2	1	15.00	15	=
19	12	12	2	0	29	5	1	1	3.00	3	=	48	50	76	5	3	78	1	1	0	17.00	17	bp
20	2	14	2	0	34	1	1	0	2.00	2	-	49	49	81	1	4	-1	0	1	0	18.00	18	stop
21	25	16	2	0	35	1	1	0	4.00	4	<	50	48	82	1	3	79	1	1	0	14.00	14	s_out
22	51	18	2	0	36	5	1	0	5.00	5	upl	51	48	83	1	3	80	1	2	0	16.00	16	t_out
23	57	-1	0	1	41	5	1	0	0.00	0	C2_	52	47	-1	0	0	81	1	1	0	0.00	0	al
24	5	20	3	1	46	1	2	0	6.00	6	%	53	12	84	2	0	82	2	1	0	3.00	3	=
25	3	23	2	1	47	1	2	0	8.00	8	*	54	47	-1	0	1	84	1	2	0	0.00	0	kl
26	12	25	2	1	48	2	2	1	9.00	9	=	55	12	86	2	1	85	2	2	1	7.00	7	=
27	3	27	3	1	50	1	1	0	6.00	6	*	56	47	-1	0	3	87	1	2	0	0.00	0	ml
28	12	30	2	1	51	2	1	0	7.00	7	=	57	12	88	2	3	88	2	2	1	13.00	13	=

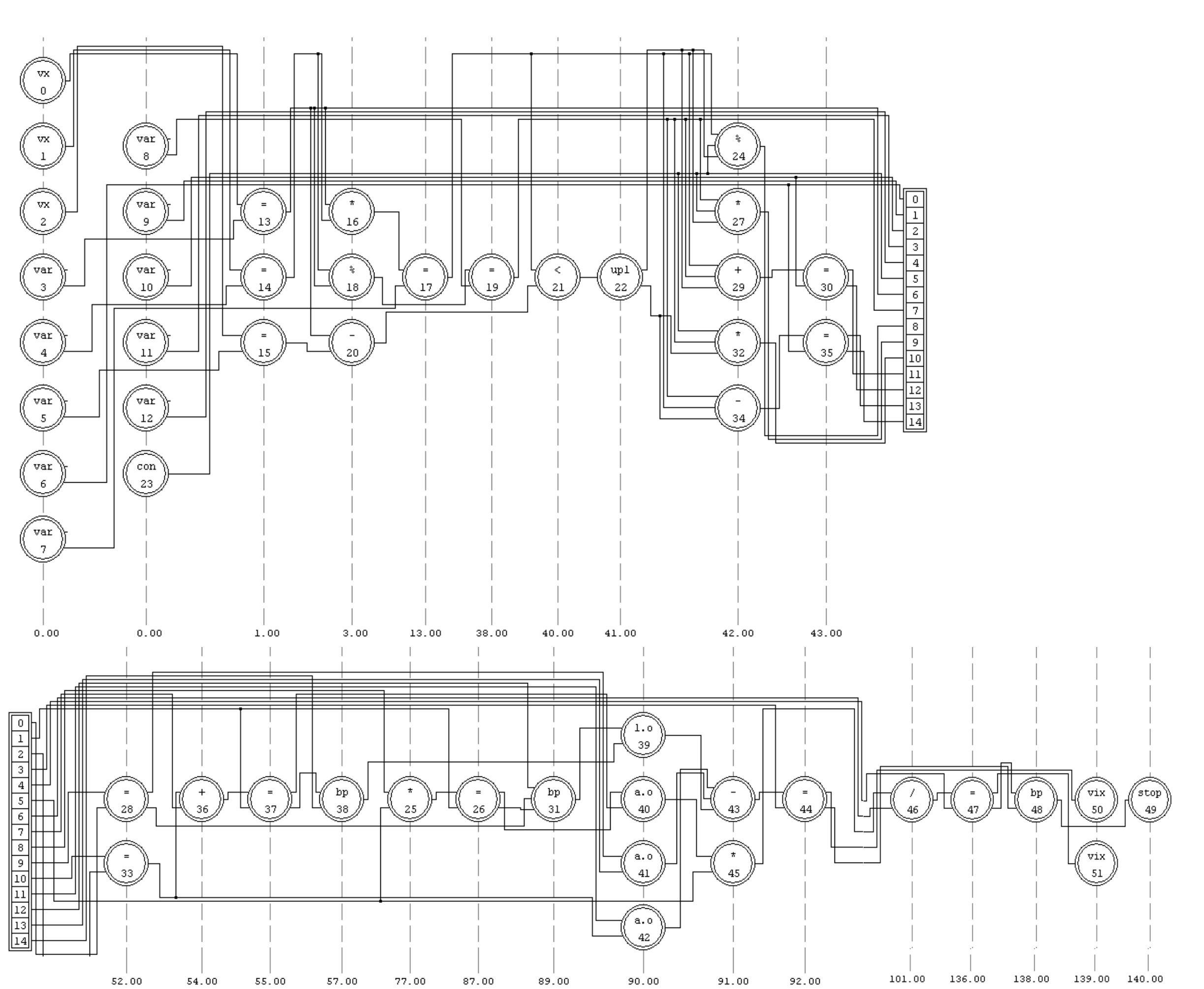

Fig. 4. Fragmentation time parameterized parallel model of C – program

The results of synthesis of texts time parameterized programs of "dedicated" processors (steps 4 ... 9) are shown in Fig. 5, 6. Parameters of operators send (1, k, 2) / receive (1, k, 2):"the source address", "value", "destination address." Placing of data in a distributed memory processor MPP computers is presented in tabl. 5.

Time parameterized program processors MPP computers with an explicit started moments NT execute the instructions (step 16) are presented in Tabl. 5, 6.

```
#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 + l;
         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);
}
```

Fig. 5. The text from the program of the first processor MPP computer (steps 8 ... 11)

```
#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);
}
```

Fig. 6. Text from the program of the second processor MPP computer (steps 8 ... 11)

Table 4

Placing data in a distributed memory of processors MPP computer (step 7)

Mem of 1-st processor		Mem of 2-nd processor	
Data addresses	Data names	Data addresses	Data names
0	b	0	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

Table 5

Processor program compute nodes MPP computers

NN	Commands 1-st processor			NT
	oper	A1	A2	
0	scanf	0	-1	1.00
1	scanf	1	-1	1.00
2	scanf	2	-1	1.00
3	*	1	0	3.00
4	=	3	-1	13.00
5	send(1,3,2)			15.00
6	-	1	2	3.00
7	=	4	-1	4.00
8	send(1,4,2)			6.00
9	%	0	1	10.00
10	=	5	-1	45.00
11	send(1,5,2)			47.00
12	<	3	4	62.00
13	upl	14	20	63.00
14	*	5	6	64.00
15	=	7	-1	74.00
16	+	3	5	64.00
17	=	8	-1	65.00
18	wait(37.00);			76.00
19	bp	27	-1	113.00
20	*	6	5	64.00
21	=	8	-1	74.00
22	send(1,8,2)			76.00
23	-	5	3	64.00
24	=	7	-1	65.00
25	wait(14.00);			80.00
26	bp	27	-1	94.00
27	-	8	7	114.00
28	=	9	-1	115.00
29	print	9	-1	117.00
30	wait(46.00);			118.00
31	bp	32	-1	164.00
32	stop	-1	-1	165.00

Table 6

Processor program compute nodes MPP computers

NN	Commands 2-nd processor			NT
	oper	A1	A2	
0	scanf	0	-1	1.00
1	wait(7.00);			3.00
2	receive(1,1,2)			10.00
3	=	1	-1	19.00

Table 6 (end)

4	receive(1,2,2)			19.00
5	=	2	-1	28.00
6	wait(21.00);			30.00
7	receive(1,3,2)			51.00
8	=	3	-1	60.00
9	<	2	1	62.00
10	upl	11	16	63.00
11	%	2	4	64.00
12	=	5	-1	99.00
13	*	5	4	101.00
14	=	6	-1	111.00
15	bp	22	-1	113.00
16	wait(16.00);			64.00
17	receive(1,7,2)			80.00
18	=	7	-1	89.00
19	+	7	3	91.00
20	=	6	-1	92.00
21	bp	22	-1	94.00
22	*	6	4	114.00
23	=	8	-1	124.00
24	/	8	0	126.00
25	=	9	-1	161.00
26	print	9	-1	163.00
27	bp	28	-1	164.00
28	stop	-1	-1	165.00

Conclusion

1. Now the problem of developing of efficient parallel software for well-known and future supercomputers has become a central issue in parallel computing techniques.

2. The developed method enables the formalization of synthesis time parallel MPP - programs based on traditional serials C-program and the known configuration of computers that meet the required specification / limitations (for a time solutions, available resources, etc.). Traditional serial and parallel programs are special cases of temporary parallel programs.

3. Described method is a basis for the creation of tools automatic synthesis of static and temporary parallel programs for well-known class of parallel computing systems MPP class and advanced Adaptive Self-assembled computing systems.

References

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

2. Лацис А. Как построить и использовать суперкомпьютер / А. Лацис. – М.: Бестселлер, 2003. – 240 с.

3. Гергель В.П. Теория и практика параллельных вычислений [Электронный ресурс] / Интернет университет информационных технологий ИНТУИТ.Р, 2009. – Режим доступа: http://www.intuit.ru/department/calculate/ paralltp/ свободный. — Загл. с экрана.

4. 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.: ХУ ПС ім. І. Кожедуба, 2008. – Вип. 3(18). – С. 118-121.

5. Поляков Г.А. Технология проектирования времяпараметризованных мультипараллельных программ как стратегия развития систем параллельного проектирования / Г.А. Поляков, Е.Г. Толстолужская // Радіоелектронні і комп'ютерні системи. – 2009. – № 6(40). – С. 166-171.

6. Поляков Г.А. Синтез и анализ параллельных процессов в адаптивных времяпараметризованных вычислительных системах: монография / Г.А. Поляков, С.И. Шматков, Е.Г. Толстолужская, Д.А. Толстолужский. – Х.: ХНУ имени В.Н. Каразина, 2012. –С. 434-575.

Поступила в редколлегию 1.11.2013

Рецензент: д-р техн. наук, проф. Г.А. Поляков, Белгородский государственный национальный исследовательский университет, Белгород.

СИНТЕЗ ЧАСУПАРАМЕТРИЗОВАНИХ ПАРАЛЕЛЬНИХ ПРОГРАМ ДЛЯ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ КЛАСУ МРР

О.Г. Толстолузька

Наведено опис методу синтезу часупараметризованих паралельних програм для обчислювальних систем з розподіленою пам'яттю (МРР). Використовувані вихідні дані: Сі - програма задачі; архітектура ЕОМ - однорідна, топологія комутаційного середовища – n - мірний гіперкуб, кількість процесорів NM, відоме число 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: *часупараметризована паралельна програма, обчислювальні системи з розподіленою пам'яттю (МРР), методи паралельної обробки даних.*

СИНТЕЗ ВРЕМЯПАРАМЕТРИЗОВАННЫХ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ КЛАССА МРР

Е.Г. Толстолужская

Приведено описание метода синтеза временных параллельных программ для вычислительных систем с распределенной памятью (МРР). 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 МРР, содержащих в явном виде время начала выполнения множеств независимых операторов.

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