

Распределение программных модулей по узлам вычислительной сети с общим полем памяти для граф-схем параллельных алгоритмов

77-30569/235842

10, октябрь 2011

Руденко Ю. М.

УДК 685.32:519.68

МГТУ им. Н.Э. Баумана

kirur@bk.ru

Как было показано в [1], весьма удобным средством для изображения параллельных алгоритмов является граф-схема. В работе [2] был предложен метод построения диаграмм решения параллельной задачи, представленной информационной граф-схемой.

Построенные диаграммы (см. рисунок 6.1.2.) является удобным средством построения плана распределения операторов по ВМ вычислительной системы. Строки временной диаграммы будем интерпретировать как нити алгоритма решаемой задачи, в соответствии с установившейся терминологией. Для удобства размещения операторов алгоритма на вычислительные модули ВС, представим структуру ВС в виде матрицы дистанций между вычислительными модулями ВС, в которой указаны расстояния между ВМ, измеренные в количестве ВМ между двумя рассматриваемыми. В качестве примера возьмём обобщённый 3-х мерный гиперкуба $2 \times 3 \times 3$, представленный на рисунке 6.2.1.

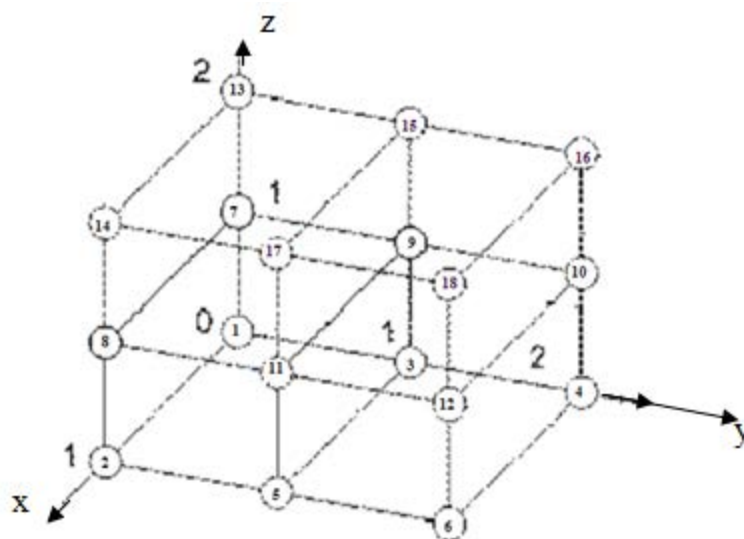


Рисунок 6.2.1. Схема представления обобщенного 3-х мерного гиперкуба $2 \times 3 \times 3$

Построим для него матрицу дистанций, которую, в дальнейшем удобно использовать для размещения нитей решаемой задачи. Нумерация ВМ производится, как показано в таблице 6.2.1.

Матрица дистанций представлена в таблице 6.2.2. В последней строке приведены итоговые суммы по столбцам. Минимальная сумма определяет лучший ВМ, который имеет минимальное расстояние ко всем остальным вычислительным модулям. В данном случае лучшим ВМ является вычислительный модуль с номерами 9, затем 11 и т. д.

Рассмотрим алгоритм, представленный граф-схемой на рисунке 6.2.2.

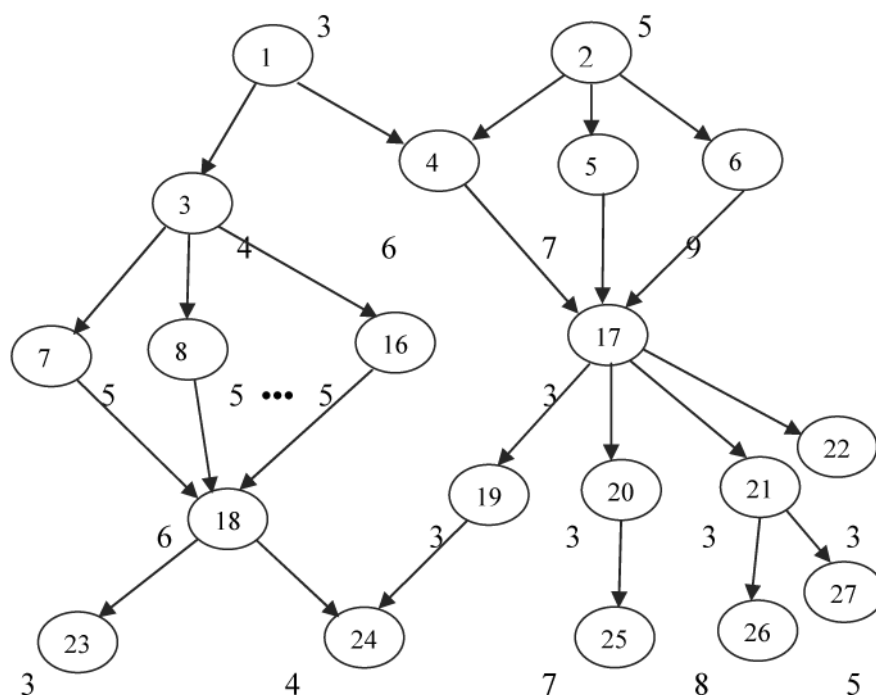


Рисунок 6.2.2. Граф-схема примера алгоритма решаемой задачи. Вершины 7, ..., 16 представляют ветви цикла. Вершина 19, ..., 22 отображает так же распараллеленный цикл

Для построения нитей решаемой задачи построим диаграмму ранних сроков окончания выполнения операторов. Используя алгоритм 6.1.1, вычислим $t_{1,1}, \dots, t_{1,27}$. Вычисленные данные представлены в таблице 6.2.3. Диаграмма ранних сроков окончания выполнения операторов представлена на рисунке 6.2.3.

Так как рассматривается ВС с общей памятью, обмена данными через каналы связи между процессорами нет, и количество нитей меньше, чем число процессоров, поэтому распределение нитей между ВМ может осуществляться, например, следующим образом: первая нить загружается в первый ВМ, вторая – во второй и т. д. Процессоры с шестого по тринадцатый освобождаются в момент относительного времени $T=18$ и могут быть распределены для решения других задач. Аналогично процессоры 6,7,8,9,10,11,12 также могут быть использованы другими задачами по мере их освобождения, что наглядно видно из рисунка 6.2.3. Стрелками на этом рисунке показаны связи между нитями.

Таблица 6.2.1

Координата ВМ(x, y, z)	000	100	010	020	110	120	001	101	011
Номер ВМ	1	2	3	4	5	6	7	8	9
Координата ВМ(x, y, z)	021	111	121	002	102	012	022	112	122
Номер ВМ	10	11	12	13	14	15	16	17	18

Построим нити для диаграммы, представленной на рисунке 6.2.2. Нити H_i составят операторы: $H_1=\{1, 3, 7, 18, 24\}$, $H_2=\{2, 8, 19, 23\}$, $H_3=\{9, 20, 26\}$, $H_4=\{10, 21, 25\}$, $H_5=\{11, 22, 25\}$, $H_6=\{12\}$, $H_6=\{12\}$, $H_7=\{13\}$, $H_8=\{14\}$, $H_9=\{15\}$, $H_{10}=\{16\}$, $H_{11}=\{4\}$, $H_{12}=\{5\}$, $H_{13}=\{6,14\}$ для $i = 1, \dots, 13$.

Таблица 6.2.2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	1	1	2	2	3	1	2	2	3	3	4	2	3	3	4	4	5
2	1	0	2	3	1	2	2	1	3	4	2	3	3	2	4	5	3	4
3	1	2	0	1	1	2	2	3	1	2	2	3	3	4	2	3	3	4
4	2	3	1	0	2	1	3	4	2	1	3	2	4	5	3	2	4	3
5	2	1	1	2	0	1	3	2	2	3	1	2	4	3	3	4	2	3
6	3	2	2	1	1	0	4	3	3	2	2	1	5	4	4	3	3	2
7	1	2	2	3	3	4	0	1	1	2	2	3	1	2	2	3	3	4
8	2	1	3	4	2	3	1	0	2	3	1	2	2	1	3	4	2	3
9	2	3	1	2	2	3	1	2	0	1	1	2	2	3	1	2	2	3
10	3	4	2	1	3	2	2	3	1	0	2	1	3	4	2	1	3	2
11	3	2	2	3	1	2	2	1	1	2	0	1	3	2	2	3	1	2
12	4	3	3	2	2	1	3	2	2	1	1	0	4	3	3	2	2	1
13	2	3	3	4	4	5	1	2	2	3	3	4	0	1	1	2	2	3
14	3	2	4	5	3	4	2	1	3	4	2	3	1	0	2	3	1	2
15	3	4	2	3	3	4	2	3	1	2	2	3	1	2	0	1	1	2
16	4	5	3	2	4	3	3	4	2	1	3	2	2	3	1	0	2	1
17	4	3	3	4	2	3	3	2	2	3	1	2	2	1	1	2	0	1
18	5	4	4	3	3	2	4	3	3	2	2	1	3	2	2	1	1	0
	45	45	39	45	39	45	39	39	33	39	33	39	45	45	39	45	39	45

Таблица 6.2.3

$t_{1,1}$	$t_{1,2}$	$t_{1,3}$	$t_{1,4}$	$t_{1,5}$	$t_{1,6}$	$t_{1,7}$	$t_{1,8}$	$t_{1,9}$	$t_{1,10}$	$t_{1,11}$	$t_{1,12}$	$t_{1,13}$	$t_{1,14}$
	5	7	11	12	14	12	12	12	12	12	12	12	12
$t_{1,15}$	$t_{1,16}$	$t_{1,17}$	$t_{1,18}$	$t_{1,19}$	$t_{1,20}$	$t_{1,21}$	$t_{1,22}$	$t_{1,23}$	$t_{1,24}$	$t_{1,25}$	$t_{1,26}$	$t_{1,27}$	
12	12	14	18	17	17	17	17	21	22	24	25	22	

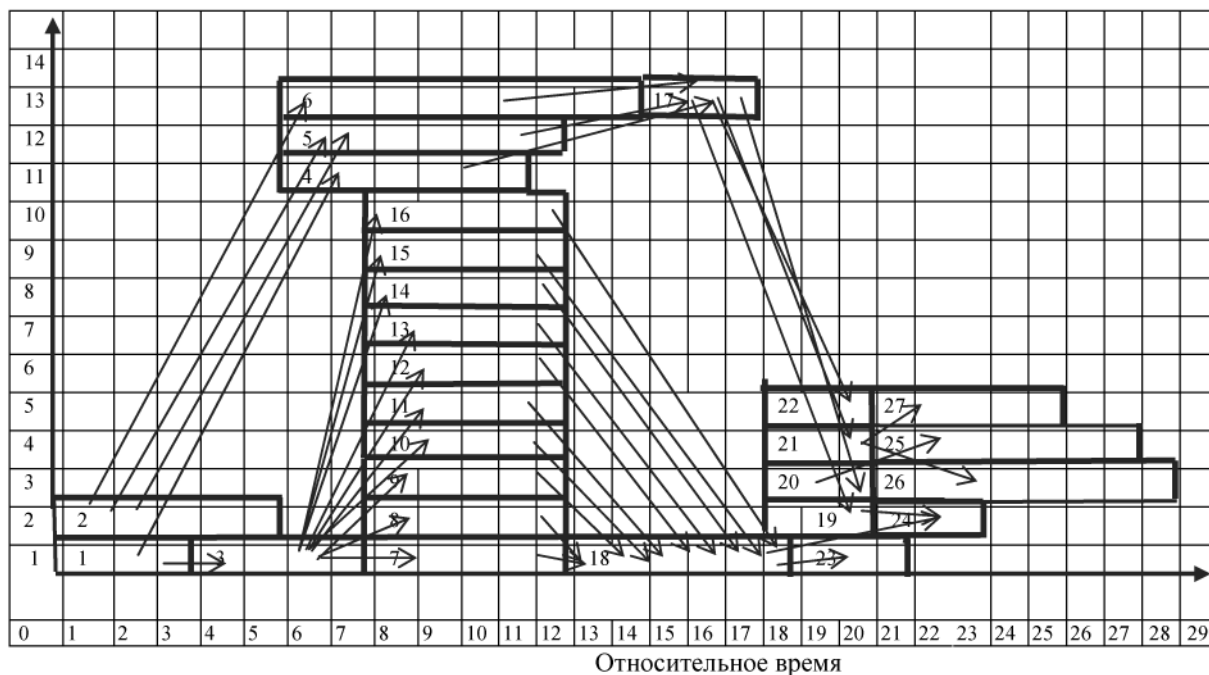


Рисунок 6.2.3. Диаграмма ранних сроков окончания выполнения операторов

Паузы, возникшие на 6, 7, 12, 13 моментах времени связаны с синхронизацией вычислений, определяемой структурой рассматриваемой граф-схемы. Общее время решения этой задачи составляет 28 условных единиц.

Анализируя представленный способ построения плана решения задач на ВС, можно сделать вывод, что достаточно простыми средствами можно построить достаточно эффективную процедуру организации решения параллельных задач на ВС.

Рассмотрим распределение операторов по ВМ вычислительной системы с общим полем памяти для информационно-логической граф-схемы

Информационно-логическая граф-схема, как известно, отличается от информационной наличием логических операторов, а так же операторов выбора вариантов, поэтому здесь возможно несколько вариантов построения планов распределения операторов по вычислительным модулям ВС. Так первый, наиболее простой, аналогичен рассмотренному в предыдущем разделе способу. Можно предположить, что логические операторы и операторы выбора вариантов условно заменены обычными, вычислительными операторами. Схема решения такой задачи аналогична выше рассмотренной.

Недостатком такого подхода является предварительное резервирование ВМ, которые не все станут востребованными. При использовании этого метода количество

неиспользуемых ВМ может быть уменьшено за счёт учёта требований на количество ВМ для обработки логических путей, использующих максимальное количество ВМ:

$$N_{\max}^{(L)} = \sum_{j=1}^r (\max_i (q_{j,i})),$$

где r – количество логических операторов в программе, $q_{j,i}$ – количество ВМ, необходимых для реализации i -й ветви j -ого логического оператора. $N_{\max}^{(L)}$ – максимальное количество ВМ, необходимых для реализации логических путей решаемой задачи. В этом случае требуемое количество ВМ уменьшится на $N^* = N - N_{\max}^{(L)}$.

Второй способ в построении некоторого динамического плана, состояние которого изменяется каждый раз, как только выполняется очередной логический оператор. Такой план в дальнейшем будем называть адаптивным. Рассмотрим построение такого плана, используя граф-схему, представленную на рисунке 6.2.4.

Диаграмма для информационно-логической граф-схемы, представленной на рисунке 6.2.4 представляется множеством диаграмм по мере поступления логических переменных. Предполагаемое количество ВМ, необходимых для решения этой задачи, можно оценить с помощью формулы $q_{j,i}$, где $q_{j,i}$ – количество ВМ, необходимых для реализации i -й ветви j -ого логического оператора. Если в граф-схеме, представленной на рисунке 6.2.4, логический оператор «один» выбирает движение по дуге 1.1, то очередная диаграмма будет выглядеть, как показано на рисунке 6.2.5.

При срабатывании условного оператора «один», дуги 1.1 требуется 10 ВМ. Время решения задачи составит 22 условные единицы.

Если же в условном операторе «один» выбирается дуга 1.2, то диаграмма преобразуется, как показано на рисунке 6.2.6. Для реализации этой диаграммы требуется всего пять ВМ, которое может быть уменьшено за счёт учёта работы логического оператора «17», Наиболее нагруженная ветвь «17.3», поэтому представим временную диаграмму для ветви «17.3. Диаграммы для остальных ветвей строятся аналогично.

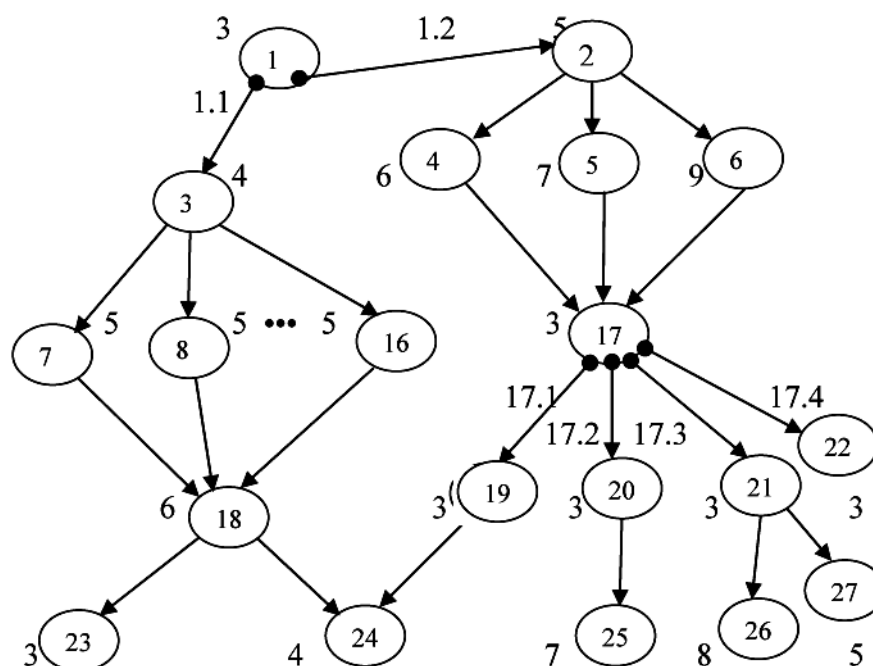
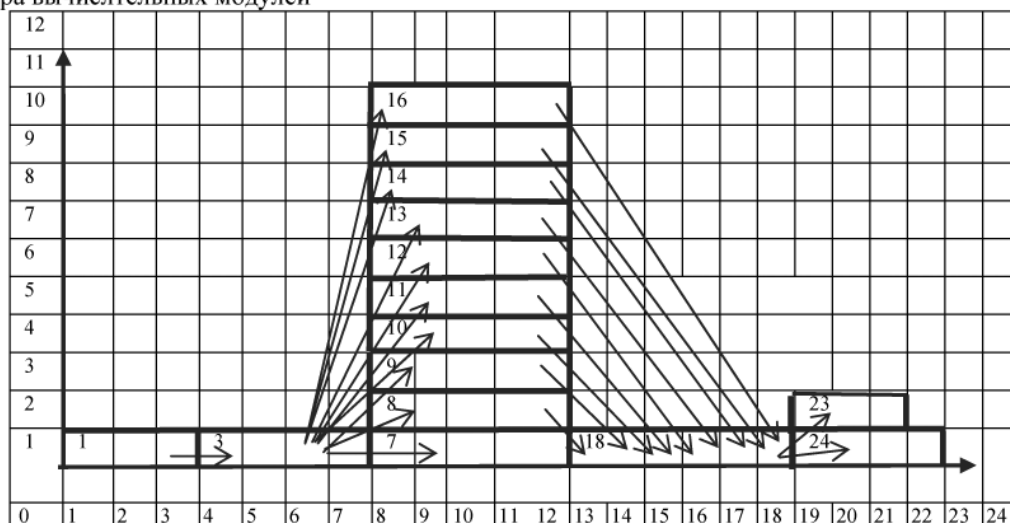


Рисунок 6.2.4. Информационно-логическая граф-схема, реализованная в виде композиций логических функций «И» и «Исключающее ИЛИ»

Номера вычислительных модулей



Относительное время

Рисунок 6.2.5. Диаграмма ранних сроков окончания выполнения операторов для информационно-логической граф-схемы при срабатывании дуги 1.1.

Номера вычислительных модулей

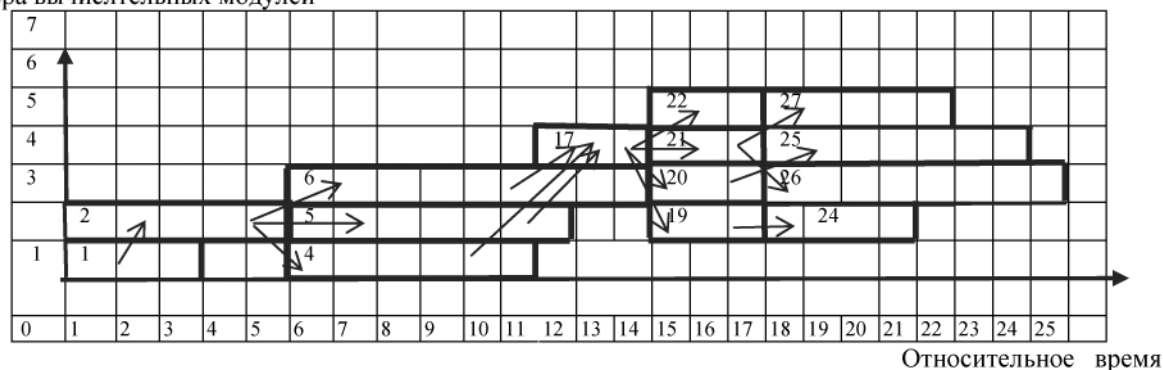


Рисунок 6.2.6. Диаграмма ранних сроков окончания выполнения операторов при срабатывании дуги 1.2.

После учёта работы всех логических операторов, в зависимости от исходных данных, определяющих ту или иную ветвь срабатывания логических операторов, количество используемых операторов колеблется от 10 до трёх. При использовании ВС с общей памятью распределение операторов по ВМ может быть произвольным. Например, первую нить, образованную операторами {1, 2, 4} можно загрузить в первый процессор, вторую, представленную множеством {5, 17, 21, 26}, выполнить на втором процессоре, и, наконец, третью нить {6, 27} выполнить на третьем процессоре. Варианты выполнения плана в случае использования функций «Исключающее ИЛИ» могут быть упорядочены в порядке убывания вероятностей. Вероятность окончания решения задачи по тому или иному варианту может быть получена на основании заранее определённых вероятностей $P_{i,j}$, где i – номер рассматриваемого логического оператора, j – номер выхода в логическом операторе. Из анализа транзитивной матрицы [3] следования информационно-логической граф-схемы решаемой задачи можно определить требуемые вероятности следующим образом. Просматривая столбцы этой матрицы, находим последние элементы столбца, содержащего конкатенацию символов, определяющих выходы операторов «Исключающее ИЛИ». Затем конкатенацию символов $i^{(r1)}_{v1_} i^{(ri)}_{vi_} \dots i^{(rn)}_{vn}$ заменяем на произведение вероятностей – $P_{peu} = P^{(r1)}_{v1_} * \dots * P^{(rn)}_{vn}$, где P_{peu} – вероятности решения задачи при том или ином варианте исходных данных решаемой задачи, $P^{(ri)}_{vi}$ – вероятность выбора gi логическим оператором vi пути.

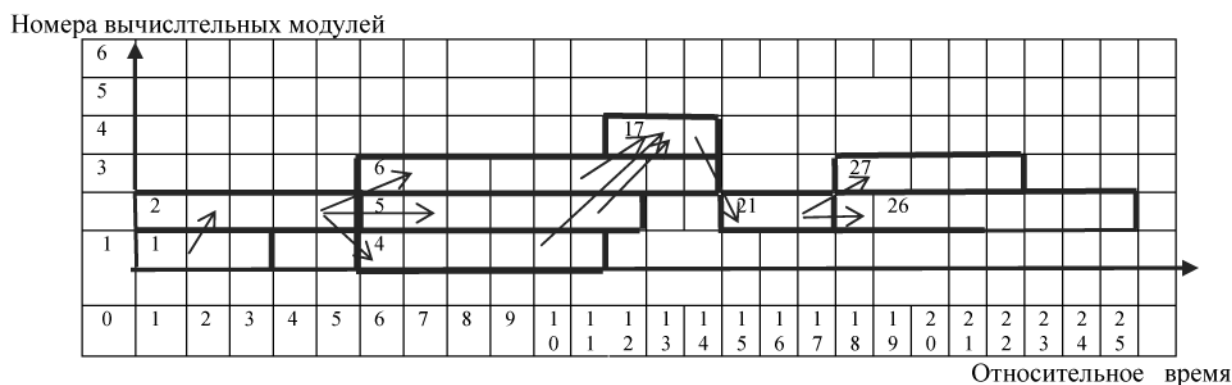


Рисунок 6.2.7. Диаграмма ранних сроков окончания выполнения операторов при срабатывании дуги 17.3.

Таким образом, предлагаемый метод построения планов распределения позволяет уменьшить время решения параллельной задачи за счёт предварительного упорядочения используемых вычислительных модулей, а также сократить их количество.

Литература

1. Руденко Ю.М. Новый подход к изображению схем алгоритмов для вычислительных систем. Информатика и системы управления в XXI веке. Сборник трудов №7 молодых учёных, аспирантов, и студентов – М.: МГТУ им. Н.Э. Баумана, 2009. 167–181 с.
2. Руденко Ю.М. Построение плана выполнения параллельных алгоритмов на базе граф-схем. Аэрокосмические технологии. Научные материалы МНТК – 2009. Реутов – Москва 2009. 179–181 с.
3. Руденко Ю.М. Учёт зависимостей программных модулей по данным и последовательностям их выполнения при параллельных вычислениях. Известия высших учебных заведений. Поволжский регион. Технические науки. № 3 (11), 2009. 67–75 с.

77-30569/235842 Distribution of software modules for unified-memory computer network components for parallel algorithm flowgraphs

77-30569/235842

10, October 2011

Rudenko Yu.M.

Bauman Moscow State Technical University

kirur@bk.ru

This paper covers a method of program module allocation into computer subsystems (CS) for the cases when the program module execution time is known. This allows you to pre-make plans to address the problem of parallel computing system, which is an important factor in developing solutions for a computer system parallel task. The authors propose creating a plan for solving the problem of information and information-logical unified-memory flowgraphs. The plan is presented in the form of parallel task flowgraph threads which a serial arrangement in computer system components. Pre-planning allocation routines for computation modules reduces the problem solution time and increases the efficiency of computing resources.

Publications with keywords: [decision](#), [a decision which is parallel to the task](#), [graph-scheme](#), [chart](#), [plan](#)

Publications with words: [decision](#), [a decision which is parallel to the task](#), [graph-scheme](#), [chart](#), [plan](#)

Reference

1. Rudenko Iu.M., Proceedings of the young scientists The collection of works of №7 of young scientists, post-graduate students and students No.7, Moscow, MGTU im. N.E. Bauman - BMSTU, 2009, pp. 167–181.
2. Rudenko Iu.M., in: Proc. of the Youth scientific and technical conference (MNTK-2009), 2009, Reutov – Moscow, 2009, pp. 179-181.
3. Rudenko Iu.M., Izvestiia vysshikh uchebnykh zavedenii. Povolzhskii region. Tekhnicheskie nauki 11 (3) (2009) 67–75.