

МЕТОДЫ РАСПАРАЛЛЕЛИВАНИЯ ПРОГРАММ ДЛЯ ПЕРСОНАЛЬНЫХ СУПЕР-ЭВМ

The article describes how the general code speedup depends on the scalar and vector concurrences. The article also describes the measurement system. This system consists of the language converter from IBM PC assembler into KADR-language, VLIW-compactor, and KADR-emulator. The speedup measurement was done in different modes for FORTRAN programs.

В работе [1] изучался вопрос о количестве крупноформатного параллелизма в задачах, характерных для персональных супер-ЭВМ (ПС ЭВМ). В настоящей статье исследуется вопрос о количестве и влиянии скалярного параллелизма на общее ускорение последовательных программ.

Поясним некоторые понятия. Под ПС ЭВМ будем, как и в [2], понимать персональную ЭВМ с традиционной последовательной системой программирования и с параллельной структурой оборудования. Главной проблемой разработки ПС ЭВМ является проблема извлечения достаточного количества параллелизма из прикладных программ.

Назовем совокупность всех АЛУ в ПС ЭВМ супер-АЛУ. Такими супер-АЛУ являются процессоры известных машин *CRAY-1*, *FPS-64*, *IBM-3090*. Как для этих машин, так и для *RISC*-процессоров разработаны распараллеливающие компиляторы следующих видов:

1. Суперскалярные (СС) компиляторы [3], которые для загрузки супер-АЛУ используют параллелизм базового блока (ББ). ББ—это линейный участок программы, не содержащий внутри себя операций перехода.

2. Векторно-суперскалярные (ВСС) компиляторы, в которых наряду со скалярным используется и векторный параллелизм [3].

3. Компиляторы на базе сверхдлинной команды [4] или *VLIW*-компиляторы (*Very Long Instruction Word*).

Отметим, что указанные компиляторы используют параллелизм внутренних циклов, а для внешних циклов используются транспьютероподобные системы и связанные с ними методы программирования.

В статье измерение количества параллелизма производится путем моделирования распараллеливающих механизмов трех указанных выше компиляторов. Тем самым решается вопрос о пригодности этих механизмов для построения ПС ЭВМ.

Согласно [4], программы можно разделить на три класса:

1. Класс *P1*—программы с большим объемом крупноформатного параллелизма в виде векторов и независимых ветвей.

2. Класс *P2*—программы, в которых параллелизм очевиден, но он слишком нерегулярный, чтобы его можно было отнести к классу *P1*.

3. Класс *S*—программы последовательной природы, например итеративные алгоритмы, точечные произведения и др.

Нетрудно предположить, что задачи классов *P2* и *S* наиболее распространены на практике и являются основными для ПС ЭВМ.

Для проведения эксперимента по измерению параллелизма были взяты семь реально используемых программ, написанных на ФОРТРАНе и относящихся к классам *P1*, *P2* и *S*:

1. *RMUL*—умножение матриц (*P1*);

2. *GAMM*—программа-тест для измерения производительности ЭВМ и трансляторов в *NFL* (*P1*);

3. *GAUSS*—решение системы линейных алгебраических уравнений методом Гаусса (*P1*);

4. *MINV*—обращение матриц (*P2*, так как в данной программе широко используется косвенная индексация, затрудняющая распараллеливание);

5. *PRQD*—вычисление корней полинома (*S*);

6. *EIGEN*—вычисление собственных значений и собственных векторов матриц (*P2*);

7. *ARAT*—аппроксимация дискретной функции методом наименьших квадратов (*S*).

Для сбора и анализа статистических данных была построена измерительная система (ИС), объем которой составляет более 50 000 строк на ЯВУ. ИС включает следующие блоки: КАСКАД (конвертор языка Ассемблер в язык КАДР); АРП (автоматический распараллеливатель); планировщик; КАДР-машину и АФП (анализатор ФОРТРАН-программ).

Входом ИС является программа на Ассемблере ПЭВМ типа *IBM PC*. Внутренним языком является язык КАДР (К-адресный язык), адекватный языкам RISC-микропроцессоров. Такое сходство языков в дальнейшем облегчит перенос ИС на конкретные микропроцессоры.

Блок КАСКАД переводит входную программу во внутреннее представление на последовательном КАДРе. Функцией КАСКАДа, кроме перевода, является также и восстановление исходного параллелизма путем увеличения количества используемых регистров общего назначения (РОН).

Автоматический распараллеливатель последовательных программ с языка КАДР на выходе вырабатывает ярусно-параллельную форму программы (ЯПФ) для каждого базового блока. Планировщик преобразует ЯПФ в параллельную программу с учетом ресурсов, имеющихся в системе (АЛУ, память, РОН и др.), разрешая все возникающие конфликты. Каждая строка программы на параллельном КАДРе содержит несколько команд, выполняемых за один такт.

Детали функционирования АРП и планировщика описаны в [4].

Анализатор ФОРТРАН-программ описан в [1] и предназначен для измерения векторного параллелизма программ, написанных на ЯВУ.

КАДР-машина эмулирует работу аппаратуры. Назначение эмулятора—отладка конвертора и упаковщика, а также сбор статистики в процессе выполнения КАДР-программ.

Основной измеряемой величиной является ускорение вычислений

$$R = T_1 / T_p,$$

где T_1 и T_p —время выполнения последовательной и параллельной программы соответственно на одно- и p -процессорной системе.

Прежде всего путем анализа статистики по ускорениям базовых блоков R_{66} было установлено, что

$$R_{66} = 1 + 0,1\omega, \quad (1)$$

где ω —число команд в ББ. Это соотношение достаточно устойчиво для языков уровня КАДР и относительно небольших ω (несколько десятков).

Таблица 1

Суперскалярное ускорение

Программа	Ускорение R_{cc} для разных p			
	1	2	3	4
<i>RMUL</i>	1,28	1,80	1,82	1,82
<i>GAMM</i>	1,12	1,38	1,39	1,39
<i>GAUSS</i>	1,33	1,83	1,84	1,84
<i>MINV</i>	1,23	1,47	1,49	1,49
<i>EIGEN</i>	1,31	1,92	2,00	2,00
<i>PRQD</i>	1,32	1,75	1,77	1,77
<i>ARAT</i>	1,44	1,96	2,05	2,05

Результаты измерения суперскалярного ускорения R_{cc} для семи программ приведены в табл. 1. Видно, что увеличение числа АЛУ в супер-АЛУ, обозначенных в табл. 1 через p , больше двух практически не приводит к увеличению ускорения. Это происходит потому, что средняя длина базового блока для прикладных программ составляет

6—12 команд и на таких базовых блоках сложно обеспечить полную загрузку более чем двух-трех АЛУ. Из (1) следует, что максимальное ускорение в этом случае должно находиться в пределах от 1,6 до 2,2.

Таблица 2

Векторное ускорение

Программа	Ускорение R_B для разных N			
	1	2	3	4
<i>RMUL</i>	1,00	1,85	3,11	4,89
<i>GAMM</i>	1,00	1,34	1,52	1,80
<i>GAUSS</i>	1,00	1,16	1,29	1,36

Измерения векторного ускорения R_B производились с помощью АФП для трех программ: *RMUL*, *GAMM* и *GAUSS*. В этих программах в векторных циклах сосредоточено соответственно 90, 50 и 35 процентов от общего объема вычислений и их можно отнести к классу $P1$. Остальные программы относятся к классам $P2$ и S и содержат мало (1—2 %) векторных вычислений. Результаты даны в табл. 2. Через N обозначено число супер-АЛУ при $n = 1$ в каждом супер-АЛУ.

Таблица 3

Векторно-суперскалярное ускорение
и ускорение *VLIW*-преобразования

Программа	Ускорение R_{BCC} для разных $N(n = 2)$			
	1	2	3	4
<i>RMUL</i>	1,80	3,34	5,84	9,14
<i>GAMM</i>	1,38	1,81	2,08	2,41
<i>GAUSS</i>	1,83	2,14	2,34	2,47
<i>MINV</i>	1,47	1,47	1,47	1,47
Ускорение R_{VLIW} для разных L				
<i>GAUSS'</i>	2,12	2,95	3,62	4,57
<i>MINV'</i>	2,15	3,25	4,68	6,12

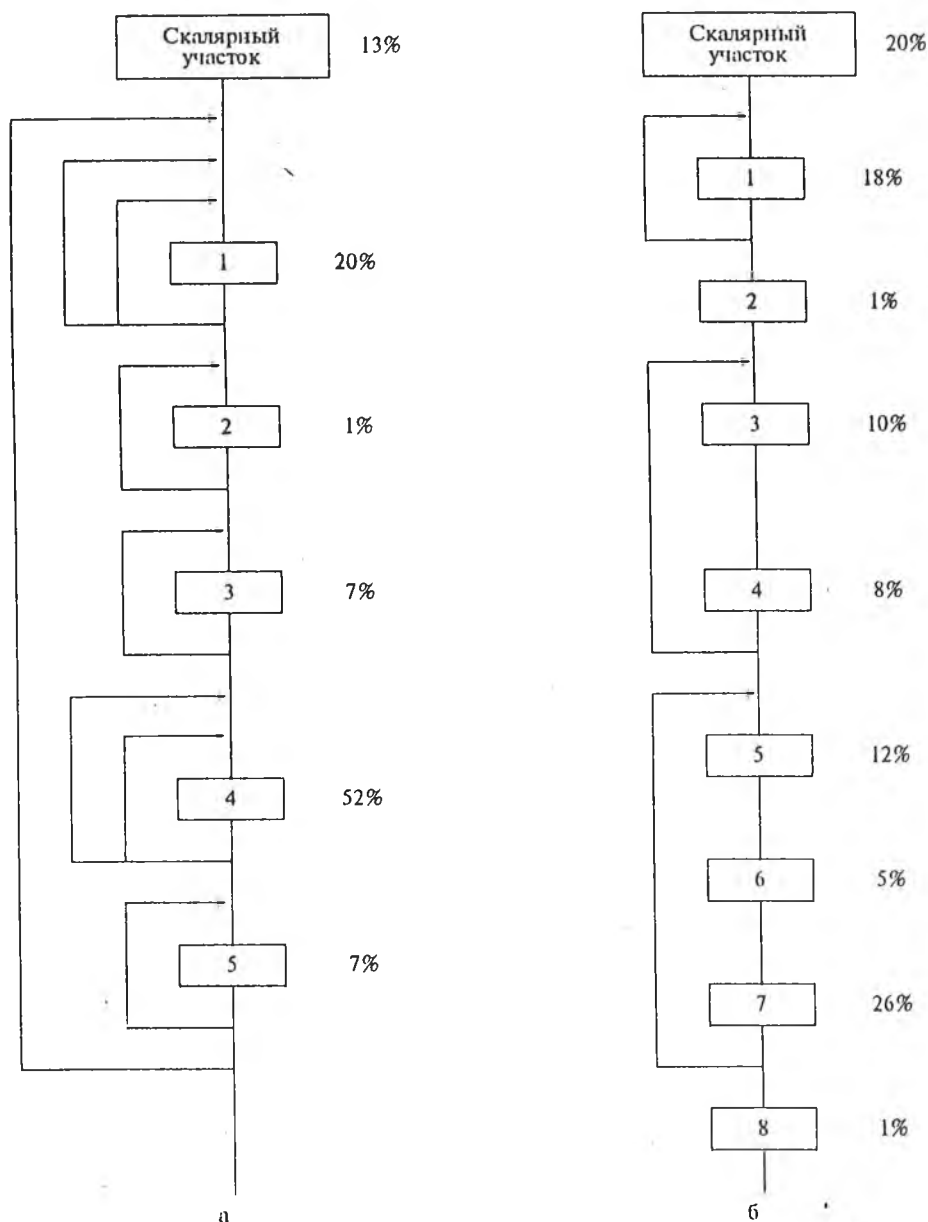
При помощи эмулятора для трех программ было измерено векторно-суперскалярное ускорение R_{BCC} (табл. 3). При этом справедливо соотношение:

$$R_{BCC} = R_{CC} * R_B.$$

Полученные результаты позволяют сделать следующий вывод: традиционный и широко распространенный механизм векторно-суперскалярной компиляции непригоден для ПС ЭВМ, поскольку вследствие слабости векторизирующих компиляторов векторный параллелизм массовых задач (класс $P2$ и S) оказывается незначительным.

Покажем, что *VLIW*-преобразование позволяет в значительной мере исключить этот недостаток. Для эксперимента возьмем две программы: программу обращения матрицы *MINV*, для которой из-за наличия в циклах операторов *IF* было получено $R_B = 1$, и программу решения системы линейных алгебраических уравнений методом Гаусса *GAUSS*.

Структура циклов этих программ показана на рисунке. Против каждого гнезда циклов проставлены цифры, соответствующие доле вычислений, заключенных в данном гнезде.



Структура циклов программ *MINV* (а) и *GAUSS* (б)

Анализ показал, что в программе *MINV* векторным является цикл 3, а циклы 1, 4, 5 не распараллеливаются из-за наличия в телах циклов операторов *IF*, а в программе *GAUSS* векторными являются циклы 4, 6, 7, а не распараллеливаются из-за *IF* циклы 1, 3, 5. Затем обе программы в соответствии с *VLIW*-преобразованием были изменены следующим образом. Тело каждого внутреннего цикла подвергается *L*-кратной развертке, затем в полученных развертках в соответствии с вероятностями переходов строятся главные и подчиненные трассы, вводятся компенсационные коды. Увеличение ББ до размеров тела цикла в соответствии с (1) увеличивает скалярный параллелизм, а развертка вовлекает в работу неиспользованный ранее векторный параллелизм.

Преобразованные таким образом программы пропускаются через АРП, планировщик и КАДР-машину. Результирующее ускорение R_{vlw} представлено строками *MINV'* и *GAUSS'* в табл. 3 и достигает достаточно большой величины.

В результате выполненной работы можно сделать предположение о перспективности использования метода VLIW для построения персональных супер-ЭВМ.

Список литературы

1. З м а ч и н с к и й С. С., Ш п а к о в с к и й Г. И., С е р и к о в а Н. В. // Программирование. 1992. № 1, С. 58.
2. Ш п а к о в с к и й Г. И. // УСМ. 1991. № 6. С. 3.
3. W i l s o n R. // Computer. Designe. 1990. September. P. 26.
4. Ш п а к о в с к и й Г. И. // ЗРЭ. 1991. № 11. С. 10.

Поступила в редакцию 28.01.93.

УДК 517.5

В. Н. РУСАК, Д. БРАЙЕСС (ФРГ)

ИНТЕГРАЛЬНЫЕ МОДУЛИ ГЛАДКОСТИ И НАИЛУЧШАЯ РАЦИОНАЛЬНАЯ АППРОКСИМАЦИЯ

The classes of 2π -periodic functions $f(x)$, which satisfy the restriction

$$\int_0^{2\pi} \left| \sum_{\nu=0}^{r+1} (-1)^\nu C_{r+1}^\nu f(x + \nu\tau) \right| dx \leq M\tau^{r+1}, \quad \forall \tau > 0,$$

are considered. The exact order of approximability of these classes by rational functions is found.

Пусть $C_{2\pi}$ — пространство непрерывных 2π -периодических функций $f(x)$, где норма, как обычно, задается равенством

$$\|f(x)\| = \sup_{-\infty < x < \infty} |f(x)|.$$

Обозначим через $W_{2\pi}^r \Omega_M$, $r = 1, 2, \dots$, классы функций из пространства $C_{2\pi}$, подчиненных ограничению:

$$\int_0^{2\pi} |\Delta_{r+1}^\tau f(x)| dx \leq M\tau^{r+1}, \quad \forall \tau > 0; \Delta_{r+1}^\tau f(x) = \sum_{\nu=0}^{r+1} (-1)^\nu C_{r+1}^\nu f(x + \nu\tau). \quad (1)$$

Пусть $E_n^T(f)$ — наилучшее приближение функции $f(x)$ в равномерной метрике тригонометрическими полиномами, т. е.

$$E_n^T(f) = \inf_{p_n(x)} \|f(x) - p_n(x)\|,$$

где $p_n(x)$ — тригонометрический полином порядка не выше n . В свою очередь, наилучшее рациональное приближение в метрике $C_{2\pi}$ определяется равенством:

$$R_n^T(f) = \inf_{r_n(x)} \|f(x) - r_n(x)\|, \quad r_n(x) = p_n(x)/q_n(x),$$

где $p_n(x)$ и $q_n(x)$ — тригонометрические полиномы порядка не выше n .

Если бесконечно малые (α_n) и (β_n) имеют одинаковый порядок, то пишем $\alpha_n \asymp \beta_n$ и через C_j обозначаем ниже положительные константы, не зависящие от n и f .

Теорема 1.

$$\sup_{f \in W_{2\pi}^r \Omega_M} R_n^T(f) \asymp \left(\frac{1}{n}\right)^{r+1}. \quad (2)$$

Доказательство соотношения (2) связано с использованием промежуточного приближения класса $W_{2\pi}^r \Omega_M$ такими функциями, которые имеют r -ые производные ограниченной вариации. Пусть $W_{2\pi}^r V_M$, $r = 1, 2, \dots$, есть класс функций $f(x)$ из пространства $C_{2\pi}$, которые представимы в виде: