

УДК 004.413.5

С.А. Резникова, А.Д. Смирнов, А.А. Чемерис

Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, Киев

СТАТИСТИЧЕСКАЯ МОДЕЛЬ ЭФФЕКТИВНОСТИ ВЫПОЛНЕНИЯ OpenMP ПРОГРАММ НА МНОГОЯДЕРНЫХ КОМПЬЮТЕРАХ

В статье представлен подход к оценке выполнения программ на параллельных вычислительных системах. Он основан на статистической информации, которая собирается при выполнении программы на данных небольшой размерности, что требует незначительных затрат машинного времени. Подход проиллюстрирован примером построения модели определения кратчайшего пути в графе.

Ключевые слова: статистическая модель, OpenMP программа, многоядерный компьютер.

Введение

Эффективность выполнения программ на различных архитектурах есть ключевым моментом для современных персональных компьютеров и вычислительных систем (ВС). Вопросами анализа алгоритмов занимаются многие коллективы. Их усилия направлены на анализ реализованных в алгоритмах методов и применения тех или иных структур данных. При этом один из основных выводов, который непосредственно вытекает из анализа взаимосвязей алгоритмов и вычислительных архитектур, состоит в том, что эффективность работы программы на конкретной вычислительной системе зависит от того, насколько соответствует реализация алгоритма соответствующей вычислительной архитектуре [1].

Однако если алгоритм уже реализован программно, то оценка его работы на конкретной вычислительной системе остается вопросом актуальным, на который еще не дан конкретный ответ, не разработаны соответствующие рекомендации. Выполнение созданной программы на многопроцессорной системе есть процесс дорогостоящий. При выполнении программы с большими массивами данных, ее работа происходит длительное время и, прежде чем запускать задачу на длительное время, требуется оценка эффективности ее выполнения. Так требуется ответить на вопросы, на скольких процессорах эффективно выполнять задачу, сколько приблизительно времени займет решение задачи для данных размером N , при каких параметрах задачи и системы следует выполнять задачу и др.

Данная статья посвящена вопросам оценки эффективности вычислений для программ, реализованных на стандарте OpenMP для вычислительных систем с общей памятью, к которым принадлежат, в частности, современные многоядерные персональные компьютеры. Для оценки эффективности работы программ предлагается использование статистических методов моделирования. С помощью такого подхода также можно оценить качество распараллеливания алгоритма и, соответственно, качество созданной параллельной программы.

Основной материал

Оценки эффективности вычислений на многопроцессорных ВС. Благодаря широкому использованию параллельных вычислительных систем, как распределенных, так и многопроцессорных, а также систем распараллеливания можно получить значительное ускорение вычислений по сравнению с однопроцессорными компьютерами. На эффективность параллельных вычислений оказывает влияние как архитектура компьютера, так и конкретная программная реализация. В теоретических рассуждениях можно предположить, что максимальная скорость вычислений равна произведению количества процессоров на максимальную скорость последовательной программы. Однако в этом определении не учтены факторы, замедляющие работу параллельного компьютера – время синхронизации, невозможность полного распараллеливания кода, особенности оборудования [2].

Наиболее простым методом определения эффективности параллельных программ может служить их оценка при помощи измерения и анализа времени. Основными метриками, которые характеризуют работу параллельных программ, является ускорение и эффективность [3].

Ускорение, получаемое при использовании параллельного алгоритма для p процессоров, по сравнению с последовательным вариантом выполнения вычислений определяется

$$S_p = \frac{T_1}{T_p},$$

т.е. как отношение времени решения задач (T_1) на скалярной ЭВМ к времени выполнения параллельного алгоритма (T_p).

Эффективность использования параллельным алгоритмом процессоров при решении задачи определяется соотношением:

$$E_p = \frac{T_1}{T_p * p} = \frac{S_p}{p},$$

т.е. величина эффективности определяет среднюю долю времени выполнения алгоритма, в течение которой процессоры реально используются для решения задачи.

Как следует из приведенных соотношений, в идеальном случае $S_p = p$ и $E_p = 1$. В идеале решение задачи на p процессорах должно выполняться в p раз быстрее, чем на одном процессоре, или/и должно позволить решить задачу с объемами данных, в p раз большими (идеальное или линейное ускорение). Ускорение называется абсолютным, когда T_1 – это время выполнения наилучшего последовательного алгоритма, а относительным – когда T_1 является временем выполнения того же алгоритма на одном процессоре.

Для исследования поведения параллельных вычислительных систем при решении задач используются модели, которые можно условно разделить на несколько классов: аналитические, имитационные и динамические, основанные на результатах трассировки выполнения задач. К аналитическим относятся такие модели как законы Амдала и Густафсона-Барсиса, модели Калахана-Эймса и Хокни и другие [3], основанные на некоторых математических зависимостях общего характера.

Имитационные модели эффективности параллельных приложений включены в ряд интегрированных программных пакетов для прототипирования, разработки и отладки параллельных программ. Также созданы пакеты, позволяющие получать симулятор различных мультипроцессоров. Например, RSIM [4] представляет собой набор инструментов, каждый из которых позволяет моделировать разные варианты отдельных функциональных компонентов многопроцессорного устройства (регистры, основная память, кэш, арифметика и т.д.). Соответствующая настройка этих компонентов позволяет получить имитатор конкретного мультипроцессора.

Модели, созданные на основе трассировки программ, позволяют учитывать влияние основных факторов, влияющих на эффективность программы – степень распараллеливания программы, равномерность загрузки процессоров во время выполнения параллельных вычислений, время, необходимое для выполнения межпроцессорных обменов, степень совмещения межпроцессорных обменов с вычислениями, а также особенности конкретной вычислительной системы. К недостаткам можно отнести то, что возможность трассировки существует далеко не во всех вычислительных системах, формат трасс не унифицирован. К тому же существует эффект замера – средство трассировки достаточно сильно меняет поведение программы.

Статистические модели. Для решения определенной выше задачи предлагается использовать

математический аппарат вероятностно-статистических методов. Его применимость обусловлена соблюдением (хотя бы приблизительно) в исследуемой реальной действительности следующих условий: а) возможностью (хотя бы мысленно реально представимой) многократного повторения наших экспериментов в одних и тех же условиях; б) наличием большого числа случайных факторов, характеризующих условия проведения экспериментов и не позволяющих делать полностью предопределенного (детерминированного) заключения о том, произойдет или не произойдет в результате этих экспериментов интересующее нас событие [5].

Модель, которая описывает исследуемый процесс, можно в общем случае представить в виде

$$Y = F(x_1, x_2, \dots, x_n), \quad (1)$$

где x_i – независимые переменные. Функция регрессии F определяет соответствие средних значений зависимой переменной Y от значений независимых переменных x_i . Решение о виде функции регрессии принимается на основе анализа статистических данных и чаще всего для построения моделей достаточно линейных функций вида

$$Y = b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k + \varepsilon, \quad (2)$$

где b_i – параметры модели, которые описывают влияние i -й переменной; ε – случайное слагаемое. Далее построение модели состоит в определении коэффициентов b_i на основании n -элементной статистической выборки.

Следующим шагом есть верификация модели, где доказывается правильность сформулированных гипотез на множестве статистических наборов данных. Выдвинутую гипотезу называют нулевой (основной) и обозначают H_0 . Конкурирующая гипотеза H_1 – это гипотеза альтернативная нулевой, т.е. противоречащая основной. Верификация осуществляется на основании значений показателей, таких как:

- R^2 – показывает уровень соответствия пространства регрессии статистическим данным; значение R^2 принадлежит диапазону $[0,1]$ и определяется выражением

$$R^2 = \frac{\sum_{z=1}^n (\hat{y}_z - \bar{y})^2}{\sum_{z=1}^n (y_z - \bar{y})^2}, \quad (3)$$

чем ближе значение R^2 к 1, тем качество модели лучше;

- коэффициент случайного изменения, который информирует о том, какой процент средней арифметической переменной модели составляет стандартное отклонение остатка; чем меньше значение этого коэффициента, тем больше соответствие построенной модели статистическим данным:

$$W_e = \frac{S_e}{\bar{y}} \cdot 100\%; \quad (4)$$

- коэффициент правдоподобия – определяет на сколько значения выходной переменной не определяется моделью; соответствие модели и статисти-

ческих данных тем лучше, чем значение коэффициента ближе к нулю:

$$\varphi^2 = \frac{\sum_{z=1}^n e_z^2}{\sum_{z=1}^n (y_z - \bar{y})^2}. \quad (5)$$

Кроме этих параметров используются и другие характеристики, например, F-статистика и t-статистика проверяют гипотезы о том, являются ли значения коэффициентов регрессии нулевыми; Q-статистика и тест Дурбина-Ватсона – характеризуют автокорреляцию, и др. Эти тесты есть таким типом тестов, которые на основании произвольной выборки статистических данных говорят о том, что принятую гипотезу следует отбросить или о том, что нет достаточных оснований для отбрасывания данной гипотезы. Кроме данных тестов, которые относятся к параметрическим тестам, существует множество непараметрических тестов, которые также используются для оценки проверяемых гипотез верификации модели [6].

Исследование OpenMP программ. Программы, созданные по стандарту OpenMP, позволяют достаточно легко внедрять параллельные конструкции в последовательные программы, а также достаточно быстро реализовывать параллельные алгоритмы. В общем виде OpenMP программа представляет собой некоторый код, в котором присутствуют специальные директивы и функции. Директивы суть указания компилятору для создания параллельно выполняющегося кода. Кроме конкретных указаний по распараллеливанию директивы содержат различные параметры, которые влияют на выполнение программ. Так, например, директива распараллеливания циклов определяет также способ распараллеливания (static, dynamic и др.), размер данных (chunksize).

В качестве примера рассмотрим выполнение программы определяющей кратчайший путь в графе между вершинами А и В, и построим модель, которая позволила бы прогнозировать время выполнения в зависимости от различных значений параметров как задачи, так и вычислительной системы. Для сбора статистической информации был произведен запуск программы для матриц размером от 100 до 1500 с шагом 100, а также для количества потоков от 2 до 28 с шагом 2. Эксперименты показали, что такие параметры как способ распараллеливания и размер данных не оказывают существенного влияния в данной программной реализации алгоритма.

На рис. 1 показана зависимость времени выполнения программы от размерности задачи. Видно, что зависимость имеет нелинейную зависимость. Программа выполнялась на персональном компьютере с 4-ядерным процессором Pentium 2,6 Ghz и оперативной памятью 4 Гб.

Регрессионная модель строилась при помощи пакета Statistica 7.0. Использовался модуль Multiple Regressions. На этапе спецификации модели были выделены независимые переменные: size – размер матрицы, Nthreads – количество потоков. В качестве зависимой переменной выбран показатель времени выполнения программы Time. После исследования корреляции переменных мы пришли к выводу, что наилучшие результаты можно получить, если искать модель вида

$$\text{Time} = b + a1*\text{size} + a2*N\text{Threads} + a3*\text{size}**2. \quad (6)$$

Результаты оценки модели на основе Multiple Regressions приведены на рис. 2, а оценки коэффициентов уравнения (6) приведены на рис. 3.

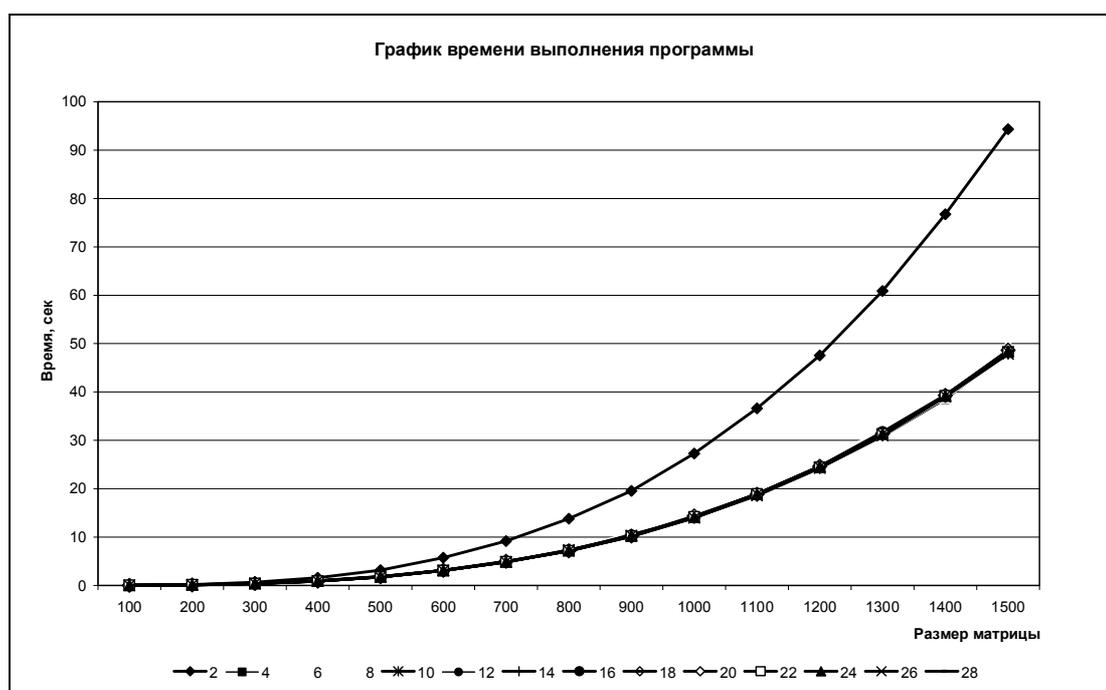


Рис. 1. Зависимость времени выполнения программы от размера матрицы связности графа при различном числе потоков в программе

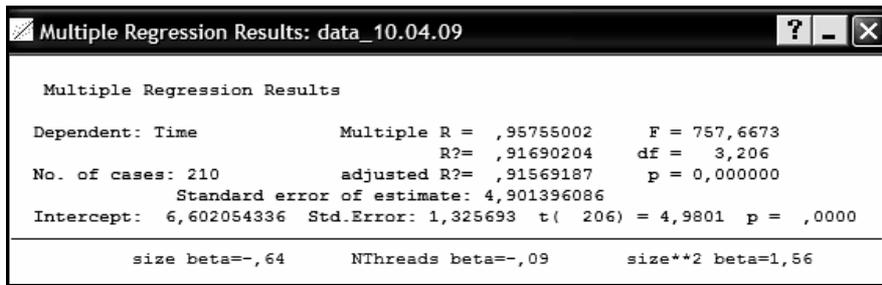


Рис. 2. Оценка модели с помощью Statistica 7.0

Regression Summary for Dependent Variable: Time (data_10.04.09)						
R= ,95755002 R2= ,91690204 Adjusted R 2= ,91569187						
F(3,206)=757,67 p<0,0000 Std.Error of estimate: 4,9014						
Include cases: 1:210						
N=210	Beta	Std.Err. of Beta	B	Std.Err. of B	t(206)	p-level
Intercept			6,602054	1,325693	4,98008	0,000001
size	-0,637458	0,086096	-0,024847	0,003356	-7,40406	0,000000
NThreads	-0,086745	0,020085	-0,181192	0,041952	-4,31901	0,000024
size**2	1,561817	0,086096	0,000037	0,000002	18,14046	0,000000

Рис. 3. Оценка коэффициентов модели

Таким образом, искомое уравнение регрессии, которое в общем виде приведено в (6), можно записать следующим образом

$$\text{Time} = 6,602054 - 0,024847 \cdot \text{size} - 0,181192 \cdot \text{NThreads} + 0,000037 \cdot \text{size}^2. \quad (7)$$

После построения уравнения линейной регрессии была проведена оценка значимости как уравнения в целом, так и отдельных его параметров. Оценка значимости уравнения регрессии в целом дается с помощью F-критерия Фишера-Снедекора. Его значение (757,67) значительно превышает критическое значение ($F(3,206) = 2,6$), а соответствующий ему уровень значимости p равен нулю, т.е. заведомо меньше 0,05. Таким образом, данная модель статистически значима.

Однако при общей значимости модели отдельные факторы могут играть разную роль, поэтому необходимо проверить и значимость каждого из факторов в отдельности в уравнении регрессии. Для проверки значимости каждого коэффициента регрессии

вычисляется t -статистика, которая показывает, во сколько раз этот коэффициент превышает свою среднюю ошибку в выборке.

Из результатов регрессии видно, что вероятность случайного появления в выборке значений t , равных или больших, чем данное значение, близка к нулю, а это означает, что все коэффициенты уравнения значимы на требуемом 5% уровне ($p\text{-level} < 0,05$).

Коэффициент детерминации определяет, что уравнение объясняет 91,69% ($R^2 = 0,9169$) вариации зависимых переменных. Это весьма высокая оценка адекватности регрессионной модели, качества уравнения регрессии (или, как говорят, мера качества подгонки регрессионной модели к наблюдаемым значениям).

На рис. 4 представлен график предсказанных и наблюдаемых значений зависимой переменной модели Time, который подтверждает адекватность модели. В табл. 1 представлены результаты прогнозирования времени выполнения программы и результаты реальных наблюдений.

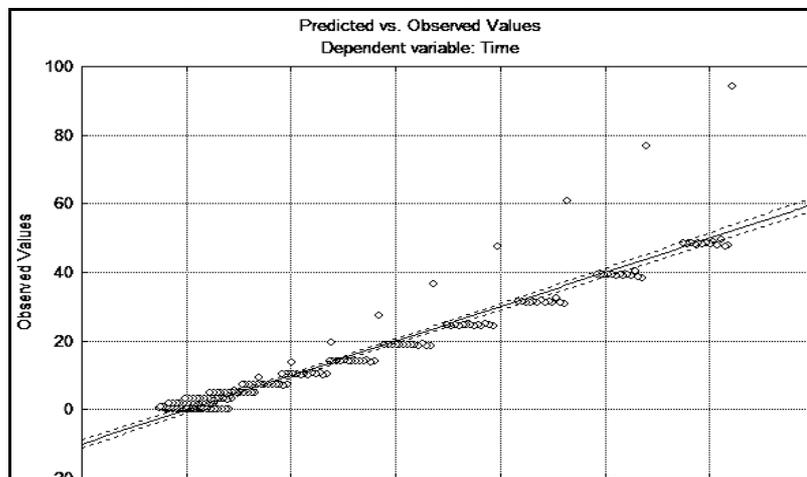


Рис. 4. Предсказанные и полученные значения

Таблица 1

Прогнозирование времени выполнения программы

№	size	Nthreads	size**2	Time observed	Time calculated	Errors
1	1700	4	2890000	69,6968	70,5614	-1,23%
2	1700	6	2890000	69,2202	70,1990	-1,39%
3	1700	8	2890000	72,6533	69,8366	4,03%
4	1700	10	2890000	70,1894	69,4742	1,03%
5	1700	12	2890000	71,5866	69,1118	3,58%
6	1700	14	2890000	70,6276	68,7494	2,73%
7	1700	16	2890000	71,0324	68,3871	3,87%
8	1700	18	2890000	70,6379	68,0247	3,84%
9	1700	20	2890000	71,0952	67,6623	5,07%
10	1700	22	2890000	71,1955	67,2999	5,79%
11	1700	24	2890000	71,0615	66,9375	6,16%
12	1700	26	2890000	70,8899	66,5751	6,48%
13	1700	28	2890000	70,9971	66,2128	7,23%

Выводы

В статье представлен подход к оценке выполнения программ на параллельных вычислительных системах. Он основан на статистической информации, которая собирается при выполнении программы на данных небольшой размерности, что требует незначительных затрат машинного времени. Подход проиллюстрирован примером построения модели определения кратчайшего пути в графе. Модель может быть использована для оценки времени выполнения программы на данной вычислительной системе.

Пример прогнозирования дает погрешность в диапазоне 1 – 7% на рассмотренном диапазоне размерностей графа, что является хорошим результатом для моделей такого класса.

Список литературы

1. Каляев А.В. Многопроцессорные системы с программируемой архитектурой / А.В. Каляев. – М.: Радио и Связь, 1984. – 240 с.

2. Столлинг В. Структурная организация и архитектура компьютерных систем: пер. с англ. / В. Столлинг. – 5-е изд. – М.: Издательский дом «Вильямс», 2002. – 896 с.

3. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – М.: БХВ, 2002. – 599 с.

4. Vijay S. Pai. Adiv RSIM: An Execution-Driven Simulator for ILP-Based Shared-Memory Multiprocessors and Uniprocessors / S. Pai Vijay, Parthasarathy Ranganathan, V. Sarita // In Proceedings of the Third Workshop on Computer Architecture Education, February 1997. IEEE Technical Committee on Computer Architecture, October 1997.

5. Айвазян С.А. Прикладная статистика и основы эконометрики / С.А. Айвазян, В.С. Мхитарян. – М.: Издательство "Юнити", 1998. – 1022 с.

6. Novatskaya E. Econometric models of some macroeconomic indicators of the Ukrainian economy / E. Novatskaya, S. Reznikova // Proceedings of the Third International Conference ACS'96. – Szczecin, Poland, November 21-22, 1996. – P. 373-380.

Поступила в редколлегию 18.03.2010

Рецензент: канд. физ.-мат. наук, с.н.с. А.А. Можаяев, Национальный технический университет «ХПИ», Харьков

**СТАТИСТИЧНА МОДЕЛЬ ЕФЕКТИВНОСТІ ВИКОНАННЯ
OpenMP ПРОГРАМ НА БАГАТОЯДЕРНИХ КОМП'ЮТЕРАХ**

С.О. Резнікова, О.Д. Смірнов, О.А. Чемеріс

У статті представлений підхід до оцінки виконання програм на паралельних обчислювальних системах. Він заснований на статистичній інформації, яка збирається при виконанні програми на даних невеликої розмірності, що вимагає незначних витрат машинного часу. Підхід проілюстрований прикладом побудови моделі визначення найкоротшого шляху в графі.

Ключові слова: статистична модель, OpenMP програма, багатоядерний комп'ютер.

**STATISTICAL MODEL OF EFFICIENCY OF IMPLEMENTATION
OF OpenMP OF PROGRAMS ON MULTINUCLEAR COMPUTERS**

S.O. Reznikova, O.D. Smirnov, O.A. Chemeris

In the article approach is presented to the estimation of implementation of the programs on the parallel computer systems. He is based on statistical information which going at implementation of the program on information of small dimension, that requires the insignificant expenses of machine time. Approach is illustrated the example of construction of model of determination of short cut in a column.

Keywords: statistical model, OpenMP program, multinuclear computer.