

Построение статистических моделей эффективности параллельных программ

В.Н.Белецкий, С.А.Резникова, А.А.Чемерис

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

В статье рассмотрен подход к оценке и прогнозированию выполнения параллельных программ на многопроцессорных вычислительных системах. Приведен пример оценки выполнения задачи матрично-векторного умножения в компьютерах с общей памятью.

В настоящее время сфера применения многопроцессорных вычислительных систем (МВС) непрерывно расширяется, охватывая все новые области в самых различных отраслях науки, бизнеса и производства. Стремительное развитие кластерных систем создает условия для использования многопроцессорной вычислительной техники в реальном секторе экономики.

Наряду с расширением области применения, по мере совершенствования МВС происходит усложнение и увеличение количества задач в областях, традиционно использующих высокопроизводительную вычислительную технику.

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. Это приводит к сокращению времени вычислений, увеличению производительности и уменьшению затрат. Очевидно, что эти показатели являются функцией от параметров вычислительной системы – числа процессоров, объема памяти, пропускной способности каналов связи и т.д. Есть также зависимость от параметров задачи, которая выполняется, а особенно важным фактором есть качество распараллеливания алгоритма задачи.

Разработка параллельных программ остается все еще сложным и трудоемким процессом. Необходимо ответить на вопрос, является ли применение МВС оправданным для конкретной задачи, при каких значениях параметров задачи и архитектуры вычислительной системы можно обеспечить максимальную эффективность использования компьютера. В идеальном случае, количество процессоров определяет, во сколько раз быстрее выполнится распараллеленная задача. Но в действительности ускорение меньше из-за организации доступа к общей памяти, необходимости синхронизации потоков, задержек в пересылках данных, а также зависимостей между переменными.

Известно, что существенным является влияние на эффективность размера программы, архитектуры вычислительных средств, количества

процессоров, способа распараллеливания, количества и размера потоков, синхронизации, операционной системы и многих других факторов.

Существующие методы исследования эффективности можно разделить на две группы:

1. Расчет на основании результатов трассировки выполнения параллельной программы. Возможен только в тех системах программирования, в которых предусмотрена возможность получения значений времени выполнения отдельных конструкций параллельной программы [1].

2. Использование моделей параллельных алгоритмов, исследующих граф алгоритма решения задач. Инструмент для создания таких моделей включается в программные пакеты разработки параллельных программ, выполняются на последовательных компьютерах и не учитывают все особенности конкретной МВС [2].

Очевидно, что предпочтительней было бы использование моделей, которые бы одновременно учитывали и характеристики алгоритма, и параметры программы, а также архитектуры, включая способ доступа к памяти и т.д. При использовании таких моделей можно было бы ответить на вопросы предварительной оценки эффективности выполнения алгоритма без запуска программы с конкретными данными и получить оценку, при каких параметрах алгоритма его выполнение будет наиболее эффективным.

Решением этой проблемы может быть использование статистических методов построения моделей, которые способны прогнозировать производительность вычислений или время выполнения программы, не запуская программу с конкретными данными. Такие модели, добавленные к программному обеспечению, помогут сделать предварительные оценки эффективности выполнения программы на конкретной ВС и подобрать параметры задачи для получения максимального быстрогодействия при решении на данной ВС.

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

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

где Y – зависимая переменная, а x_i – независимые переменные.

Решение о виде функции регрессии принимается на основе анализа статистических данных и чаще всего для построения моделей достаточно линейных функций вида

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

Правильность построенной модели в значительной мере зависит от соответствующего множества переменных, которые имеют соответствующее влияние на исследуемый параметр. Переменные, используемые в модели должны:

- 1) иметь возможность изменения в широком диапазоне;
- 2) иметь сильную корреляцию с исследуемой переменной;
- 3) иметь слабую корреляцию с другими независимыми переменными.

“Оптимальный” состав факторов, включаемых в статистическую модель, является одним из основных условий ее “хорошего” качества. Проблема выбора “оптимальных” факторов обычно решается на основе содержательного и количественного (статистического) анализа тенденций рассматриваемых процессов.

На этапе содержательного анализа решается вопрос о целесообразности включения в модель тех или иных факторов, исходя из “здорового” смысла.

На этапе статистического анализа решается вопрос о том, насколько те или другие независимые переменные влияют на значение независимой переменной, какие из них нужно оставить в модели, а какие исключить.

При построении модели мы будем использовать апостериорный подход к уточнению состава факторов статистической модели. Таким образом, на этом этапе построения модели определяется множество переменных, входящих в выражение (1). Далее производится оценка коэффициентов b_i ($i=0,1,\dots,k$) выражения (2). Для этого применяют известные статистические методы, например, метод наименьших квадратов

Для иллюстрации и доказательства применимости данного подхода для анализа эффективности алгоритмов распараллеливания приведем пример, в котором умножается матрица на вектор. В данном примере оцениваем время выполнения программы для разной размерности задачи при разном количестве потоков.

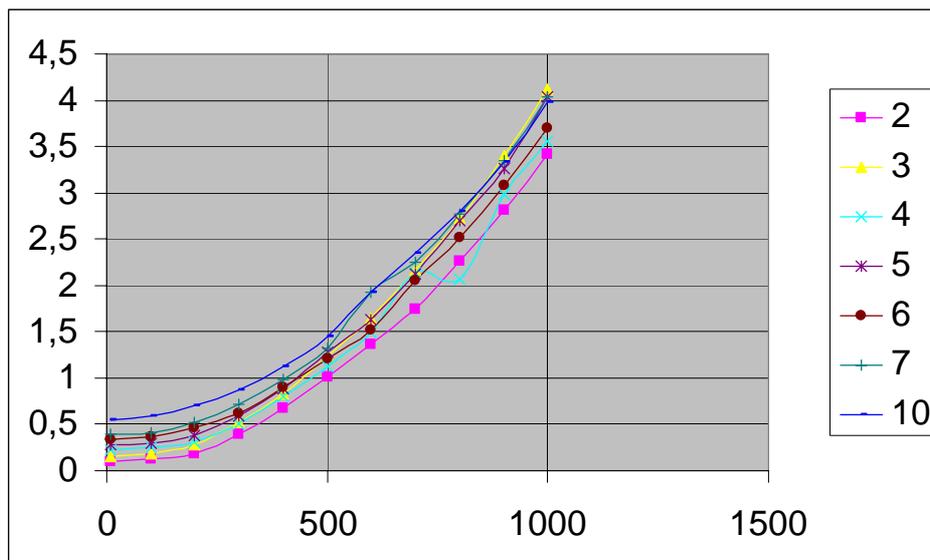


Рисунок 1

На рис. 1 приведены результаты тестовых выполнений задачи при различных значениях размерности задачи N и числа потоков программы P . При этом статистические данные получены при выполнении программы для размерностей задачи в диапазоне 10 – 1000 уравнений с шагом 100. Эти данные отображены по оси x на рис. 1. Программа запускалась соответственно как 1, 2, 3, 4, 5, 6, 7 и 10 потоков. По оси y отображены значения времени выполнения программы для соответствующих N и P . Статистика выполнения программы получена на двухпроцессорном компьютере с архитектурой SMP, который построен на процессорах Pentium III/800 MHz.

Анализ полученных статистических данных, построение и верификация модели осуществлялась в программе STATISTICA, который содержит модули автоматического определения коэффициентов модели, а также генерирует оценки правдоподобия модели. Для нашей задачи была построена регрессионная модель $T=f(N,P)$, где T – время решения задачи, в виде

$$T = b_0 + b_1 \cdot N + b_2 \cdot P . \quad (3)$$

На основе полученных числовых значений модель принимает вид

$$T = -0,542 + 0,0036 \cdot N + 0,0546 \cdot P \quad (4)$$

Квадрат коэффициента детерминации $R^2=0,0299$ свидетельствует о хорошем соответствии линии регрессии и экспериментальных данных.

Полученную модель можно использовать для прогнозирования времени вычисления данной задачи при различных значениях размерности.

Таблица 1

Размерность задачи	Количество потоков	Время, рассчитанное на основании модели, сек	Время, полученное на основании эксперимента, сек
1100	4	3,638	3,078
1200	5	4,052	4,015
1400	6	4,827	4,922

Как видно из табл. 1 построенная модель может быть использована для оценки времени выполнения программы при выбранных значениях независимых переменных.

Выводы

Анализ эффективности параллельных вычислений для оценивания получаемого ускорения вычислений и степени использования всех возможностей компьютерного оборудования при параллельных способах решения задач - один из важных аспектов использования МВС.

Использование статистических методов для анализа времени выполнения алгоритмов на параллельных компьютерах является актуальной задачей. При этом могут быть получены ответы как на вопросы эффективности выполнения алгоритмов на конкретной вычислительной системе, так и на вопросы соответствия выполняемого алгоритма и данной параллельной ВС. Основным достоинством применения такого подхода есть возможность прогнозирования эффективности выполнения программ без необходимости ее выполнения на МВС.

Литература

1. Коновалов Н.А., Крюков В.А., Погребцов А.А., Сазанов Ю.Л. C-DVM – язык разработки мобильных параллельных программ, Москва: изд. ИПМ им. М.В.Келдыша РАН, 1997.
2. Wolfe M., High Performance Compilers for Parallel Computing, Addison-Wesley Publishing Company, 1995.

Дата надходження до редакції 07.12.2007 р.