УДК 004

Р. А. ХИСАМУТДИНОВ

ПРОБЛЕМЫ АДЕКВАТНОСТИ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И КРИТЕРИИ РАЗРАБОТКИ НОВОГО ПОКОЛЕНИЯ ТЕСТОВЫХ СИСТЕМ

Рассмотрены проблемы оценочного тестирования вычислительных систем. Описаны некоторые распространенные тесты и тестовые системы и предложен новый подход к оценке производительности многопроцессорных систем.

Оценка производительности; *быстродействие*; *тестовые программы*

Производительность средств вычислительной техники, т. е. способность произвести определенный объем вычислений за единицу времени, является одной из ключевых характеристик компьютера, а оценочное тестирование с целью определения производительности всегда было одной из актуальнейших задач производителей вычислительной техники и профессиональных потребителей, а также культовой задачей многочисленных пользователей.

В общем случае, под оценкой производительности предполагается получение неких числовых значений, позволяющих ранжировать вычислительные системы и (или) оценить временные затраты на решение задач. На практике же выясняется, что существующие методы оценки производительности, как правило, прогнозируют эффективность выполнения на данной системе задач определенного класса.

Такая постановка позволяет, с одной стороны, подбирать для решения задачи вычислительные системы определенной конфигурации, а с другой стороны, проектировать вычислительные системы, оптимизированные по производительности на определенном классе задач.

Важность применения оценочного тестирования значительно возросла в последнее время, когда стало возможным и экономически выгодным строить системы из готовых компонентов: вычислительных узлов на базе материнских плат персональных машин или рабочих станций и высокоскоростных коммутационных сетей [1].

Существует еще одна задача оценки производительности — организация учета высокопроизводительных систем. В настоящее время это организуется с помощью известного теста Linpack, но намечается тенденция к переходу на более сложный набор тестов HPC Challenge benchmark [2].

В статье рассматриваются известные способы оценки производительности, их достоинства и недостатки и предлагаются новые критерии тестирования многопроцессорных систем, позволяющие наиболее адекватно оценивать производительность на различных режимах работы основных функциональных узлов.

ПРОБЛЕМЫ ОЦЕНОЧНОГО ТЕСТИРОВАНИЯ СКАЛЯРНЫХ ОДНОПРОЦЕССОРНЫХ КОМПЬЮТЕРОВ

В настоящее время существует множество вариантов тестовых программ (бенчмарок — benchmark), предназначенных для оценки производительности.

В общем случае, задача оценки производительности не является тривиальной. Если ограничиваться измерением быстродействия конкретных узлов, то можно легко получить искомое значение. Сложности возникают при выборе базисной команды, время исполнения которой и будет измеряться. Рассмотрим вариант простейшей реализации фоннеймановской архитектуры — компьютеры, не имеющие конвейерной обработки и кэш-памяти, т. е. устройств, без которых современные процессоры немыслимы. Этому типу примерно соответствуют компьютеры типа DEC PDP-11, IBM PC, PCjr, XT. Производительность процессора можно измерить как скорость выполнения:

- регистрового сложения (целочисленного);

• регистрового умножения (целочисленного);

• умножения чисел (вещественных)

или любым из множества других операций. Арифметико-логические устройства таких процессоров имеют определенную длительность исполняемых операций, измеряемую количеством необходимых тактовых импульсов. При этом время выполнения одной операции $T_{\text{оп}}$ будет определяться как

$$T_{\text{оп}} = \frac{N_T}{F},$$

где N_T — количество тактов, необходимое для выполнения операции; F — тактовая частота (Гц).

Это справедливо для скалярных процессоров с одним функциональным блоком. При рассмотрении скалярных процессоров, имеющих несколько функциональных блоков, за время $T_{\text{оп}}$ может быть произведено несколько операций. Если производить оценку производительности по операции регистрового сложения (что на большинстве современных процессоров происходит за один такт), то процессоры с несколькими функциональными узлами (например, Alpha 21264) за один такт могут произвести несколько подобных операций.

Но «узким» местом в производительности отдельных скалярных процессоров являются не арифметико-логические устройства или блоки операций с плавающей запятой, а шины доступа к памяти. Оценивая производительность скалярного процессора по теоретически возможному количеству операций (так называемая *пиковая производительность*), мы не учитываем поступления команд в процессор, а также обращения к памяти за данными и запись результатов в память, что, несомненно, обязательно в любой реальной программе. Если процессор имеет сложную систему команд (архитектуру CISC), определение производительности как возможности исполнить определенное количество операций за единицу времени становится неадекватным из-за значительной разницы во времени исполнения различных команд, особенно использующих сложные методы адресации операндов. В RISC-системах эта проблема проявляется не столь значительно в силу особенностей системы команд и отсутствия сложных способов адресации в арифметических командах.

При тестировании системы с кэш-памятью данных время исполнения команды с выборкой данных из памяти может отличаться на 1–2 порядка в зависимости от того, попадает ли обращение в кэш или нет. Системы, построенные на процессорах с поддержкой множества асинхронных взаимодействующих легких процессов (мультитредовые системы, multithread [3]), еще более непредсказуемы в силу более эффективной загрузки функциональных блоков процессора.

Суммируя вышесказанное, можно выделить следующие проблемы оценки производительности однопроцессорных вычислительных систем:

1) Невозможность выбора команды, время исполнения которой будет принято за эталонное при оценке.

2) Невозможность выбора эталонного метода адресации операндов для адекватной оценки времени подготовки данных.

3) Невозможность однозначной оценки времени доступа к данным вследствие возможного наличия многоуровневой кэш-памяти.

4) Усложнение архитектуры процессора, системы команд, микроалгоритмов предварительной выборки и исполнения инструкций, что приводит к невозможности назначения определенного числа как характеристики производительности системы.

Очевидно, что адекватная характеристика производительности может быть выработана только условно в двух граничных состояниях. Производительность не может превысить пиковую производительность, т. е. производительность по самой быстрой команде без учета выборки и подготовки операндов, а также записи результата (как правило, это команда регистрового сложения). Оценка «снизу» может быть произведена по самой «тяжелой» команде, имеющей наибольшее время выполнения с учетом подготовки и выборки операндов из памяти с наиболее сложным методом косвенной адресации, а также сохраняющей результат операции в память (если это допустимо системой команд процессора); при этом подразумевается, что попаданий в кэш нет. Имея такие крайние оценки, отличающиеся, как правило, на несколько порядков, возможно оценить время исполнения программы, но тоже с аналогичным разбросом.

ПРОБЛЕМЫ ОЦЕНОЧНОГО ТЕСТИРОВАНИЯ СКАЛЯРНЫХ МНОГОПРОЦЕССОРНЫХ КОМПЬЮТЕРОВ

В настоящее время подавляющее большинство многопроцессорных систем создается на базе готовых, коммерчески доступ-

ных процессоров. Такую систему на аппаратном уровне можно представить как совокупность вычислительных модулей и коммуникационной среды. Проблемы, возникающие при оценке однопроцессорных систем, в данном случае не теряют своей актуальности, но к ним добавляется еще ряд неоднозначностей, связанных с функционированием коммуникационной среды и межпроцессным взаимодействием.

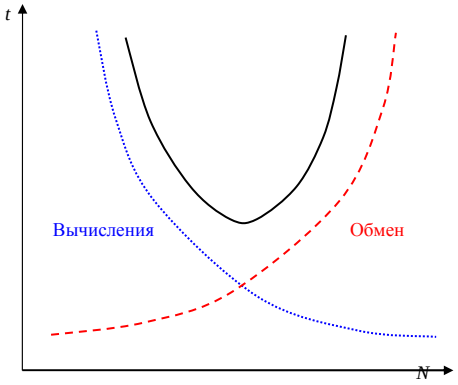

Рис. 1

На рис. 1 изображены графики зависимости времен вычислений, обмена и исполнения некоей условной задачи от количества используемых процессоров. Разделяя последовательную задачу на параллельные подпрограммы (гранулы) и распределяя эти гранулы на отдельные процессоры, мы получаем заметное уменьшение времени вычислений. Это не линейный график в силу того, что в каждую гранулу добавляются операции, обеспечивающие взаимодействие гранул. В предельном случае размер каждой гранулы составит 1 команду и время вычислений сведется к некоему минимуму, обусловленному необходимостью обеспечения последовательности обработки (предполагается, что число процессоров в системе не менее числа гранул). Но с увеличением числа гранул возрастает время, необходимое для передачи данных между гранулами. Чем меньше гранула, тем больший объем информации подлежит приему/передаче в масштабах всей системы. Безусловно, существует класс задач, в которых увеличения обмена при возрастании степени параллелизма не происходит — например, это некоторые задачи криптографии. Но в общем случае рост интенсивности межпроцессных обменов — неизбежная плата за распараллеливание. Общее время исполнения задачи представляет сумму времени вычислений и времени обмена, что приводит к наличию точки экстремума на результирующем графике. На реальных задачах физический смысл такого минимума — наличие оптимального для сочетания {программа} + {вычислительная_система} количества используемых процессоров.

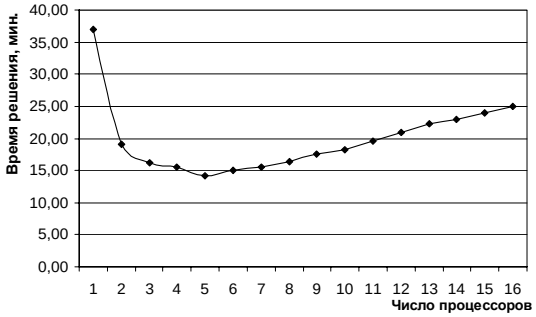

Рис. 2

График на рис. 2 показывает зависимость времени исполнения пакета Eclipse100 с реальным набором данных от количества задействованных процессоров в кластере.

Приведенные графики заставляют иначе взглянуть на проблему оценки производительности — из них следует, что на некоторых реальных задачах возрастание пиковой производительности системы, определяемой как сумма значений пиковой производительности всех процессоров в системе, не приводит к росту производительности системы в целом. Объяснение этого факта становится очевидным на следующем графике (рис. 3).

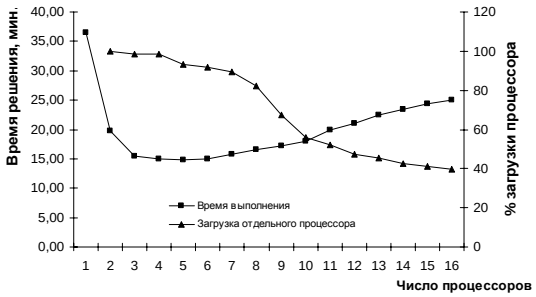

Рис. 3

При малом числе процессоров вычислительная нагрузка на каждый процессор велика и близка к 100%. Увеличение числа процессоров, т. е. увеличение числа гранул, позволяет снизить эту нагрузку, но при этом процессоры начинают простаивать в ожидании данных от других процессоров. Поступление данных задерживается из-за характеристик коммуникационной среды. С ростом интенсивности обмена происходит увеличение задержек, что приводит к простою процессоров и, как следствие, увеличению вре-

мени исполнения программы. Можно предположить, что увеличение масштаба системы и степени распараллеливания приведет к тому, что на множестве процессоров задача будет выполняться большее время, чем на одном процессоре. Улучшая характеристики коммуникационной среды, можно сдвинуть точку минимума вправо, а увеличивая производительность каждого процессора при неизменной коммуникационной среде — влево. Для другой задачи, с меньшей интенсивностью обменов, оптимальное количество процессоров, обеспечивающее минимальное время вычислений, будет иным, возможно, большим, чем для рассмотренной.

В результате такого анализа становится очевидным, что характеристика производительности многопроцессорной вычислительной системы сильно зависит от реализуемого алгоритма, что делает выработку одного числового значения как характеристики производительности невозможным.

ТЕСТОВЫЕ ПРОГРАММЫ

К настоящему времени разработано множество программ для оценки производительности вычислительной системы. Эти программы принято называть бенчмарками. Как правило, результатом такого тестирования является число, показывающее один из возможных вариантов:

- производительность системы в определенных единицах. Некоторые тесты учитывают реальное количество выполненных операций, другие исходят из теоретического минимума операций определенного типа для решения данной задачи;
- производительность системы в единицах относительно эталонной вычислительной системы.

Как правило, подобные тестовые программы представляют либо реальные вычислительные задачи или их фрагменты (тесты-ядра), или являются смесью команд определенного типа (синтетические бенчмарки).

Подавляющее число бенчмарок относится к так называемому классу тестовых программ с неизменяемой нагрузкой. Они содержат определенное количество команд, зависящее, возможно, от размерности обрабатываемых данных (если имеется возможность изменения этой размерности). Наиболее известный тест такого типа — Linpack, представляющий собой пакет для решения системы линейных уравнений методом LU-факторизации. В параллельном варианте тест базируется на распространенных пакетах SCALAPACK и BLAS. Размерность матрицы (так же, как и другие параметры) задается в конфигурационных файлах. По результатам теста с июня 1993 г. ведется известный рейтинг 500 наиболее производительных вычислительных систем мира []. Ключевые параметры списка — $R_{\max}$ — производительность по Linpack и R_{peak} — пиковая производительность в операциях с плавающей запятой. Определение производительности по Linpack производится следующим образом:

1) Количество произведенных в процессе решения системы уравнений размерностью N операций N_{op} определяется как теоретически необходимое количество вычислений,

$$N_{\text{op}} = \frac{2}{3}N^3 + \frac{3}{2}N^2.$$

2) Измеряется длительность временного промежутка между началом вычислений (после генерации матриц) и окончанием расчетов (перед выводом результатов).

Отношение этих двух величин и есть «производительность по Linpack». Очевидно, что все дополнительные команды, имеющиеся в программе, такие как вызовы подпрограмм и возвраты из них, пересылка данных и прочие, не учитываются.

Тесты с изменяемой нагрузкой предполагают увеличение объема обрабатываемых данных в процессе тестирования. Наиболее известный тест HINT вычисляет определенный интеграл функции $y = \frac{1-x}{1+x}$ в диапазоне [0,1]. При этом диапазон разбивается на участки. На каждом участке точность вычисления увеличивается. При этом используется метод иерархической интеграции как функции используемой памяти. Результатом работы теста является функция производительности в единицах QUIPS (Quality Improvements per Second — качественные усовершенствования в секунду) либо от времени работы теста, либо от используемой в процессе счета памяти. При увеличении точности требуется выполнять больше итераций, следовательно, необходимо больше памяти для хранения различных численных данных и индексов. Сначала (при небольшом объеме вычислений) тест довольствуется памятью первичного кеша. При увеличении объема вычислений приходится задействовать вторичный кеш, основную память, а при необходимости — и дисковую память. Таким образом, время выполнения для одиночной системы складывается из времени выполнения команд процессором и времени, связанного с выборкой памяти.

На рис. 4 показан пример результата исполнения теста HINT [4]. Необходимо отметить, что это один из немногих тестов, который представляет результат в подобном виде, что позволяет видеть производительность при обработке данных определенной размерности.

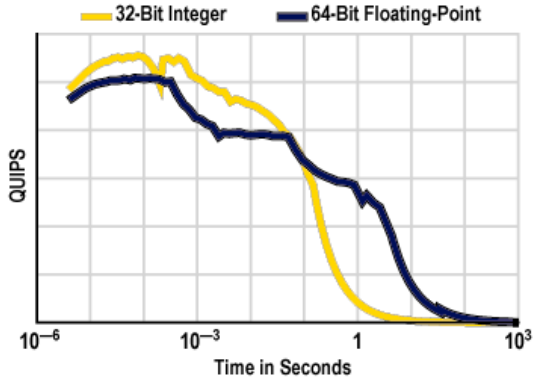

Рис. 4

ТЕСТОВЫЕ КОМПЛЕКСЫ

Как говорилось выше, используемые для оценки производительности бенчмарки в ряде случаев представляют собой реальные программы решения распространенных задач. В связи с тем, что вычислительная система, как правило, имеет различные характеристики производительности для различных алгоритмов (или для различных смесей команд), адекватная оценка производительности требует наличия широкого спектра тестовых задач. Это решается с помощью тестовых программных комплексов, содержащих несколько программ, реализующих наиболее распространенные алгоритмы.

Для однопроцессорных вычислительных систем широкое распространение получили так называемые тесты SPEC — комплексы программ, разработанных Standard Performance Evaluation Corporation (SPEC, [5]), в которую входит около 60 различных поставщиков средств вычислительной техники. Один из наиболее известных и в то же время новых тестовых программных комплексов — SPEC CPU2000. Этот программный комплекс предназначен для определения вычислительной мощности и не затрагивает производительности дисковой подсистемы, подсистемы ввода-вывода, графической подсистемы.

Тест CPU2000 состоит из двух частей. Для измерения производительности на целочисленных операциях используется группа CINT2000, состоящая из 12 программ. Состав теста CINT2000 приведен в [6]. Операции с вещественными числами более трудоемки (по микрокоду), чем операции с целыми числами, поэтому для определения производительности на числах с плавающей запятой требуются свои тесты. В тестах SPEC CPU2000 для этого используется группа из 14 программ CFP2000. Результат выполнения в тестах SPEC обычно представляется в формате *speed*, что позволяет оценить быстродействие в процентах относительно базовой системы (Sun Ultra 10).

Для оценки производительности многопроцессорных систем в последнее время все шире применяется тест HPCC (High Performance Computer Challenge benchmark).

Последняя версия этого теста включает 7 групп программ:

1) Linpack — производительность на решении системы линейных уравнений;

2) DGEMM — производительность на задачах матричных умножений с двойной точностью;

3) STREAM — анализ устойчивости и пропускной способности ОЗУ на задачах векторных вычислений с данными, объем которых превосходит размер КЭШа [7];

4) PTRAMS — скорость связи пары процессоров, эффективность их взаимодействия, общая коммуникационная характеристика системы [8];

5) RandomAccess — скорость случайных обновлений данных в оперативной памяти;

6) FFTE — скорость вычислений с двойной точностью на задачах дискретного преобразования Фурье [9];

7) Набор тестов для исследования времени отклика процессоров, а также ширины пропускания коммуникационной сети. Используются тесты b_eff [10].

ИНТЕГРАЛЬНАЯ ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

Обобщая описания методик тестирования и проблем оценочного тестирования, можно сделать следующие выводы:

1) Существующий подход к оценке производительности (независимо от типа бенчмарки) показывает быстродействие вычислительной системы на смеси команд. Характеристика производительности соответствует только этой смеси команд и может иметь совершенно другие показатели при изменении алгоритма.

2) Изменение нагрузки в тестах позволяет отследить изменения в производительности системы при увеличении объема обрабатываемых данных, но также только при определенной последовательности команд. Применение иного алгоритма может привести к изменению показателей производительности.

3) Тесты не показывают изменения характеристик производительности системы при изменении числа задействованных процессоров.

Использование тест-ядер не может решить все проблемы. По своей сути, подобные программы способны оценить степень пригодности вычислительной системы к решению определенного алгоритма. Является неправильной трактовка об оценке такими бенчмарками характеристик системы на *классе задач*. Фактически речь идет об оценке одного метода (алгоритма) решения задачи, а количество подобных методов может быть большим. Переход к тестовым пакетам, в которых собраны программы оценки различных характеристик вычислительной системы, не позволяет вывести интегральную оценку производительности, так как для пользователя (реального или потенциального) важна информация не о том, как быстро осуществляется доступ к памяти или решается система линейных уравнений. Пользователю необходимо увидеть, сколько будет решаться его задача, содержащая, в основном, команды типа A, обрабатывающие объем данных B, при распараллеливании на C гранул при предполагаемой интенсивности межпроцессных обменов D.

В связи с этим наиболее эффективным представляется создание нового класса синтетических тестов, которые будут соответствовать следующим условиям для адекватной интегральной оценки производительности:

1) Возрастание нагрузки в широком диапазоне. Это позволит оценить работу системы при различных объемах обрабатываемых данных с учетом иерархии памяти.

2) Изменение нагрузки на устройства выборки и дешифрации команд с учетом конвейерной обработки и качественного изменения смеси команд.

3) Изменение производительности системы при возрастании интенсивности и объема межпроцессорных обменов.

4) Изменение производительности системы при изменении количества используемых процессоров.

Такая постановка приводит к 5-мерной характеристике производительности вычислительной системы и позволит адекватно оценивать вычислительную мощность, достижимую при исполнении алгоритма с известными характеристиками. Безусловно, получение корректного результата зависит от корректности постановки условий алгоритма. Определение таких характеристик алгоритма как интенсивность и объем обмена, степень параллелизма, преобладающий тип используемых операций, является нетривиальной задачей, но позволит произвести правильную оценку пригодности алгоритма к исполнению на вычислительной системе заданной конфигурации.

СПИСОК ЛИТЕРАТУРЫ

1. **Эйсымонт, Л. К.** Оценочное тестирование высокопроизводительных систем: цели, методы, результаты и выводы : сб. лекций 2-й Всерос. молодежн. шк. «Суперкомпьютерные вычислительно-информационные технологии в физических и химических исследованиях» / Л. К. Эйсымонт. Черноголовка, 2000.
2. **http:**//icl.cs.utk.edu/hpcc/ [Электронный ресурс].
3. **Norton S. J.** Thread Time: Multithreaded Programming Guide / S. J. Norton, M. D. Dipasquale [by Hewlett-Packard Company]. Prentice Hall PTR, 1997.
4. **http:**//hint.byu.edu/documentation/Gus/HICSS98/HICSS98.html [Электронный ресурс].
5. **http:**//www.spec.org. [Электронный ресурс].
6. **http:**//www.ixbt.com/cpu/ insidespeccpu2000.shtml. [Электронный ресурс].
7. **http:**//www.cs.virginia.edu/stream/. [Электронный ресурс].
8. **http:**//www.netlib.org/parkbench/. [Электронный ресурс].
9. **http:**//www.ffte.jp. [Электронный ресурс].
10. **http:**//www.hlrs.de/organization/par/services/ models/mpi/b_eff/. [Электронный ресурс].