Оценка быстродействия нерегулярного доступа к памяти

Дмитрий Волков, Александр Фролов

Расширение пропасти между производительностью процессоров и скоростью доступа к памяти, появление приложений, интенсивно взаимодей?ствующих с памятью через единое адресное пространство, стимулировали создание вычислительных систем с новой архитектурой. Однако для оценки таких систем традиционные тесты уже не подходят. Пришло время тестов «анти-Linpack».

Известно, что скорость выполнения многих приложений, работающих на высокопроизводительных системах, заметно отстает от их заявленной пиковой производительности, что объясняется, в частности, неоптимальной организацией работы с памятью и коммуникационной сетью. Аналогия здесь исключительно проста: неважно, с какой скоростью может работать конвейер по сборке автомобилей, если ему вовремя не будут поставлены необходимые узлы— значит, он будет простаивать. Высокая производительность, демонстрируемая процессорами компьютерной системы и фиксируемая на тестах Linpack или SPEC, оказывается по этой причине малоинформативной— реальная производительность определяется задержками при работе с памятью и сетью. Данная проблема неадекватности оценки только ухудшается. Производительность процессоров ежегодно растет на 60%, но задержки при обращении к памяти снижаются лишь на 7%, а для коммуникационных сетей ситуация сложнее, но, в целом, еще хуже. Сегодня задержки обращения к памяти оцениваются в сотни тактов процессора, а к сети— от нескольких тысяч до десятков тысяч тактов.

Пока речь идет об обычных приложениях с хорошей пространственно-временной локализацией обращений к участкам памяти небольшого объема, проблему больших задержек можно решить путем увеличения объемов кэш-памяти и использования статического (Itanium) или динамического (Pentium) суперскалярного параллелизма выполнения соседних машинных команд. Однако для ***приложений с интенсивным и нерегулярным доступом (Data Intensive Systems, DIS)*** к большим участкам памяти эти приемы не помогут. Требуются новые архитектуры и решения.

Обычно проблема становится нагляднее и быстрее решается при наличии соответствующих оценочных тестов. Например, рейтинг Top500, составленный на базе результатов теста Linpack, способствовал прогрессу в достижении высокой производительности процессоров. Применяемые реализации этого теста имеют лучшую из известных пространственно-временную локализацию обращений к памяти. Для решения проблемы повышения эффективности подсистемы памяти применим тест RandomAccess, оценивающий работу памяти в наихудшем режиме ее использования с самой плохой пространственно-временной локализацией обращений к памяти большого объема по случайным адресам. Этот тест оценивает эффективность такого доступа в единицах измерения, называемых GUPS (Giga-Updates per Second— миллиард операций модификации памяти, выполненных за секунду), поэтому его иногда упрощенно называют тестом GUPS. Другое название этого теста, «анти-Linpack», отражает тот факт, что он с иной стороны, нежели Linpack, оценивает вычислительную систему.

DIS-приложения

К категории DIS относят приложения со следующими общими свойствами:

- низкая повторяемость обращений к одним и тем же участкам памяти (низкая временная локализация);
- малая вероятность обращений к памяти по последовательным адресам (низкая

пространственная локализация);
- большой объем памяти, необходимой для работы приложений (высокая интенсивность работы с данными);
- слабая предсказуемость смены адресов участков памяти, к которым происходят обращения.

Эти свойства делают бесполезными усилия по увеличению объемов кэш-памяти и применению схем «предварительной накачки» данных. Одна задача может и не обладать всеми этими свойствами, однако тест RandomAccess имеет все эти свойства сразу, причем каждое свойство в наихудшем варианте. К перечисленным свойствам DIS-приложений недавно добавились еще требования выполнять на фоне вычислений такого типа обработку колоссальных потоков данных. Это направление получило название Data-Intensive Computing (DIC).

В качестве примеров DIS-приложений можно назвать: задачи, оперирующие с разреженными матрицами; задачи на графах; расчетные задачи на динамически изменяемых сетках нерегулярной структуры; задачи дискретного имитационного моделирования; коммерческие приложения, интенсивно работающие с корпоративными базами данных. Впервые данная аббревиатура была упомянута в проекте DARPA DIS, в рамках которого был составлен пакет тестовых задач и проведены исследования для оценки эффективности решения таких приложений, как радиолокация, обработка и распознавание сложных изображений и сигналов [1]. К настоящему времени к числу тестовых задач добавились обработка разведывательной информации, создание биологического оружия нового поколения, системы управления боевых роботов. Среди «мирных» примеров подобных приложений можно назвать: криптоанализ; средства наблюдения и слежения; моделирование сложных физических систем, включая разработку новых источников энергии и безопасных энергетических установок; задачи из области нанотехнологий; прогнозирование погоды и климатических изменений; разработка новых лекарственных препаратов; информационно-управляющие социально-экономические системы.

Компьютерные системы для выполнения подобных приложений активно ведутся сейчас в рамках проектов DARPA HPCS и DARPA PCA (разработка наземных и бортовых компьютеров), отечественного проекта «Ангара», целью которого является создание суперкомпьютера стратегического назначения, а также специального комплексного проекта Тихоокеанской северо-западной национальной лаборатории Министерства энергетики США по практическому внедрению вычислительных систем нового поколения [2].

Проиллюстрируем на примере, как традиционные системы справляются с задачами DIS. На рис. 1 для микропроцессора Itanium 2/1,3 ГГц с пиковой производительностью 5,2 GFLOPS приведены полученные в НИЦЭВТ данные для теста из набора EUROBEN (поэлементное умножение двух векторов, www.euroben.nl). Четыре варианта этого теста отличаются способом выборки элементов векторов из памяти. В случае последовательной выборки векторов с единичным шагом результаты самые высокие, но и они составляют лишь 23% от пиковой производительности, а при усложнении выборки результат ухудшается, вплоть до 6% при выборке элементов векторов по индекс-вектору. При этом на рис. 1 приведены данные для коротких векторов, которые могут поместиться в кэш-памяти, но если длины векторов увеличить, то реальная производительность на этом тесте может упасть до 0,3% от пиковой.

Данный тест показывает, насколько чувствителен микропроцессор Itanium 2 к пространственно-временной локализации данных и интенсивности работы с памятью. Аналогичные результаты демонстрируют микропроцессоры Pentium, Opteron и Power.

Комплект HPC Challenge

Сегодня при оценочном тестировании вычислительных систем уже невозможно пользоваться одним-двумя тестами для анализа производительности— требуется система из нескольких специально подобранных тестов. Разработанный в рамках программы DARPA

HPCS пакет оценочных тестов HPC Challenge— яркий пример такого комплекта.

HPC Challenge содержит семь тестов:

- HPL является одной из реализаций теста Linpack, измеряющего производительность вычислительной системы при решении систем линейных уравнений;
- DGEMM показывает производительность при умножении плотнозаполненных матриц;
- STREAM используется для измерения пропускной способности памяти при регулярном доступе;
- RandomAccess измеряет скорость выполнения обращений к памяти по нерегулярным адресам;
- FFT показывает производительность системы при выполнении быстрого преобразования Фурье;
- PTRANS измеряет пропускную способность сети, наблюдаемую при параллельном транспонировании матриц;
- b_eff используется для измерения задержек и пропускной способности сети при обмене сообщениями между узлами.

Тесты в HPC Challenge подобраны таким образом, чтобы исследовать производительность системы на всем диапазоне пространственно-временной локализации обращений к памяти. Так, например, тесты HPL и DGEMM имеют высокую пространственно-временную локализацию, FFT— высокую временную и низкую пространственную локализацию, а STREAM— низкую временную и высокую пространственную локализацию при единичном шаге по памяти. На рис. 2 показан пример диаграммы сравнения на этом наборе тестов трех систем из 64 узлов на базе микропроцессоров AMD Opteron, использующих разные коммуникационные сети (RapidArray, Quadrics и Gigabit Ethernet).

Для перспективных вычислительных систем нового поколения, предназначенных для выполнения DIS-приложений, наибольшее значение имеет составляющая G-RandomAccess теста HPC Challenge.

Тест RandomAccess

Тест RandomAccess (icl.cs.utk.edu/projectsfiles/hpcc/RandomAccess) состоит в выполнении операций модификации ячеек памяти, расположенных по случайным адресам в пределах приблизительно половины физически доступной памяти в исследуемой системе. Под модификацией памяти подразумеваются три операции над 64-разрядным словом: чтение его из памяти, выполнение над ним какой-либо арифметико-логической операции (сложение, логическое И, логическое ИЛИ, исключающее ИЛИ) и запись слова обратно в память. По аналогии с оценкой производительности вычислительных устройств, оцениваемых в MFLOPS, принята единица измерения производительности по тесту RandomAccess— 1 GUPS.

Эта программа тестирования случайным образом читает и обновляет отдельные блоки памяти, имитируя процесс доступа приложений к данным, например, для заполнения большой таблицы или построения гистограммы путем «просеивания» большого объема исходных данных. Элементы памяти индексируются в виде большого массива (при этом способ разбиения адресов между процессорами в системе с распределенной памятью не специфицируется), а номер элемента подбираются с помощью генератора случайных чисел LSFR, чьи параметры вычисляются так, чтобы обеспечить максимальный период повторяемости. Ясно, что для систем с кэш-памятью индексированное чтение по произвольным адресам— достаточно тяжелая операция, сводящая на нет все усилия разработчиков по увеличению кэш-памяти и обычно применяемая для улучшения показателей при выполнении традиционных тестов. Для получения приемлемых результатов по тесту RandomAccess кэш-память процессора должна быть равна всей доступной оперативной памяти системы.

Для разработчиков и пользователей представляет интерес GUPS-рейтинг исследуемой системы и рейтинг ее отдельных компонентов, например, GUPS-рейтинг многопроцессорной системы с распределенной памятью, GUPS-рейтинг SMP-узла или отдельного процессора. Результаты выполнения тестов RandomAccess публикуются на сайте HPC Challenge (icl.cs.utk.edu/hpcc/hpcc_results.cgi) и имеют официальный статус. Кроме того, этот тест в обязательном порядке используется во множестве исследовательских работ по архитектуре, при испытаниях нового оборудования. Максимальный результат на тестах RandomAccess в начале 2008 года составлял 35,5 GUPS для суперкомпьютера BlueGene/L и 33,6 GUPS для Cray XT3. Однако у создаваемых по проекту DARPA HPCS перспективных систем этот показатель должен быть 64000 GUPS.

Каждому суперкомпьютеру— по GUPS

В таблице 1 приведены результаты тестирования некоторых систем.

Лучший результат по G-RandomAccess на системе IBM BG/L был получен за счет использования большого количества процессоров (65536)— несмотря на то, что отдельный процессор PowerPC440 оказывается в два с половиной раза хуже, чем AMD Opteron в системе Cray XT3 (последняя заняла второе место при использовании меньшего количества процессоров— 25600).

Показатель S-Random для современных коммерческих динамически суперскалярных микропроцессоров до недавнего времени не превышал 0,015 GUPS для одного ядра, при этом наблюдалась деградация при увеличении используемого количества ядер и вычислительных узлов. Показатель S-Random для современных статически суперскалярных микропроцессоров (VLIW или EPIC архитектура) находится на уровне 0,002 GUPS (например, SGI Altix 4700 и другие системы с микропроцессором Itanium 2), что почти в восемь раз хуже, чем для динамически суперскалярных микропроцессоров. Рекордный результат для одного процессора составляет 0,59 GUPS и получен на векторном заказном микропроцессоре системы NEC SX-8.

Как видно из таблицы, показатель G-Random на современных вычислительных кластерах на базе суперскалярных микропроцессоров и коммерческих коммуникационных сетей масштабируется плохо. Типичная ситуация— HP XC 3000, для которого на 64 ярах (32 процессора Woodcrest) удается увеличить производительность в три раза до 0,045 GUPS, однако далее идет деградация. Показатель G-random плохо масштабируется и для векторных процессоров, исключение составляет лишь Cray X1E.

Ситуация с результатами на тесте RandomAccess меняется весьма динамично. Данные по кластерам на январь 2008 года уже гораздо лучше; во всяком случае, для кластеров Intel Atlantis, Discovery и Endeavour на 1000 процессорах был получен показатель 2,5 GUPS. Векторные процессоры также демонстрируют рост показателей и улучшение масштабируемости. Например, на экспериментальной установке с опытными кристаллами системы Cray BlackWidow уже на 64 процессорах получено значение 5,4 GUPS; ожидается, что на серийных кристаллах этот показатель будет вдвое выше. Однако, современные кластеры на базе коммерчески доступных микропроцессоров имеют все-таки низкую производительность в GUPS, поэтому в некоторых системах выполняют своеобразный «тюнинг» путем добавления внекристальных специализированных СБИС, которые реализуют функции работы с глобально адресуемой памятью и при этом обеспечение толерантности к задержкам. Такой тьюнинг был сделан в Cray T3E и Cray XT3/XT4/XT5, планируется в Cray Cascade за счет добавления сетевого микропроцессора GEMINI. Умельцы придумали еще один путь повышения показателя производительности за счет программной реализации функций «тьюнинговых» внекристальных СБИС, выполняемых на дополнительных ядрах микропроцессора. Такое решение выглядит достаточно разумным, учитывая, что вычислительная мощность этих ядер на задачах с интенсивным и нерегулярным доступом к данным бесполезна. В этом направлении уже получены обнадеживающие результаты [3].

***

Тест RandomAccess, как и тест Linpack, уже начал играть свою положительную роль, определяя направление усовершенствования вычислительных систем— повышаются характеристики классических систем, появились новые системы, которые можно объединить в рейтинг по этому тесту. Явный фаворит в этой гонке— Cray XMT (Eldorado) на базе 8 тыс. процессоров, для которого предполагается получить показатель в 120 GUPS. Один из первых образцов Cray XMT планируется установить в Тихоокеанской северо-западной национальной лаборатории для выполнения исследований и разработок по проекту DARPA DIC, а образцы других фаворитов— Cray BlackWidow и CrayBaker будут развернуты в Окриджской национальной лаборатории.

Литература

1. Data Intensive Systems (DIS) Benchmark Performance Summary. AFRL-IF-TR-2003-198. Final Technical Report, August 2003. Titan Corporation.
2. Data Intensive Computing. Pacific Northwest National Laboratory. U.S.Department of Energy. 2007.
3. K. Underwood, M. Levenharden, R. Brightwell, Evaluating NIC Hardware Requirements to Achive High Message Rate PGAS Support on Multi-Core Processors. SC07, November 10-16, 2007.

Дмитрий Волков— сотрудник ИПМ им. М. В. Келдыша РАН, Александр Фролов (frolov@nicevt.ru)— начальник сектора программного обеспечения ОАО НИЦЭВТ (Москва).

Программа создания перспективных суперкомпьютеров
http://www.osp.ru/os/2007/09/4566841

Технология RandomAccess

Тестирование осуществляется на массиве T размером 2^n 64-разрядных слов, над которыми выполняется модификация на основе логической операции исключающего ИЛИ: T[k] = T[k] XOR a_i, где a_i— это 64-разрядное слово из потока A_i псевдослучайных чисел, генерируемых полиномом ($x^{63} + x^2 + x + 1$). Изначально задается, что количество обращений к элементам массива T в четыре раза больше количества его элементов, что дает шанс элементам T попасть в кэш-память.

При выполнении теста имеется ряд дополнительных условий, например, возможно отложенное исполнение модификаций, но каждому процессору разрешается сохранять не более 1024 неисполненных модификаций. В случае параллельного исполнения модификаций возможна ситуация, при которой одновременно модифицируется одна и та же ячейка памяти; в этом случае одна модификация будет потеряна, что допускается, однако количество потерянных модификаций не должно превышать 1%.

Тест Random-Access предлагается запускать в трех вариантах.

- S-RandomAccess— обработка осуществляется локально только на одном узле или процессоре.
- EP-RandomAccess. Имеется несколько узлов, каждый из которых выполняет последовательный тест S-RandomAccess без взаимодействия между узлами (вариант получил название Embarassingly Parallel— «неограниченно параллельный»).
- G-RandomAccess. Глобальное выполнение, при котором несколько узлов вместе обрабатывают один тест над общим массивом данных T, распределенным между узлами. Поток Ai делится между процессорами на интервалы, так что каждый процессор

работает только со своим собственным интервалом.

Имеются две версии реализации— локальная (последовательная) и глобальная (параллельная, на MPI). Последовательная при использовании соответствующих компиляторов может быть векторизована или распараллелена с использованием тредов. Имеется вариант теста на языке UPC, в котором массив T находится в адресуемой тредами памяти, а элементы его циклически распределены между тредами. Параллельная версия написана с использованием MPI-1 для двух случаев— число процессоров является степенью двойки или не является. В последнем случае не удается распределить память поровну между процессорами, поэтому для MPI-программ, построенных по SIMD-модели вычислений, происходит рассинхронизация, приводящая к снижению показателей теста.

Процедура сборки теста достаточно стандартна— требуется запустить скрипт для Unix-утилиты make(1), причем имеются варианты для большинства системных платформ.

HPC: факторы влияния
http://www.osp.ru/os/2007/10/4705518

03.03.2008г.
