УДК 004.432

СИСТЕМА ТЕСТОВ ДЛЯ ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК ПРОИЗВОДИТЕЛЬНОСТИ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

А. В. Адинец[1], **П. А. Швец**[2]

Предложена система тестов для графических ускорителей. Созданные тесты позволяют измерять такие характеристики, как задержка и пропускная способность различных типов памяти, эффективность атомарных операций и размер кэш-линии. При помощи разработанного специального теста памяти показано, что когерентность оказывает значительно большее влияние на пропускную способность памяти, чем локальность. Статья рекомендована к печати Программным комитетом Международной суперкомпьютерной конференции “Научный сервис в сети Интернет: экзафлопсное будущее” (http://agora.guru.ru/abrau2011).

Ключевые слова: графические процессорные устройства, неоднородные вычислительные системы, атомарные операции.

Введение. В последнее время часто для решения вычислительных задач используются программируемые ускорители, в число которых входят, например, графические процессорные устройства (ГПУ), процессоры CELL BE [1] и процессоры ClearSpeed [2]. Одномашинные и кластерные системы на их основе в совокупности составляют класс неоднородных (гетерогенных) вычислительных систем. Наиболее перспективными в настоящее время представляются гетерогенные системы на основе ГПУ, что определяет актуальность задачи исследования и анализа производительности программ для таких систем, выявлению в них узких мест, определению потенциала для оптимизации и тонкой настройки.

Достаточно типичной является ситуация, когда пользователь написал программу для ГПУ, запустил ее на гетерогенном кластере, а обещанного 100-кратного ускорения не получил. Такое происходит по целому ряду причин. Данные могли быть неудачно расположены в памяти, в результате обращения к данным соседних потоков не объединялись в одну транзакцию и скорость чтения падала на порядок. Или же могло быть запущено слишком мало потоков и их было недостаточно для сокрытия задержки доступа. Или соседние потоки могли расходиться на условных операторах. Или на ГПУ все хорошо, но передача данных с хоста и обратно съедает все ресурсы. Такой список можно продолжать еще достаточно долго. Обо всех подобных узких местах и причинах низкой производительности хотелось бы знать, чтобы иметь возможность их обходить. Кроме того, надо иметь стандартные примеры, которые показывали бы возможности по устранению таких узких мест, если они встречаются в конкретных программах.

Архитектура графических ускорителей сильно отличается от традиционных процессоров, и для написания высокоэффективных приложений необходимо не только понимать архитектуру, но и уметь предсказывать эффективность разных программных конструкций с учетом особенностей работы графического ускорителя. Некоторые характеристики и особенности прямо указываются производителями графических ускорителей. Однако другие, значения которых нельзя получить через стандартные интерфейсы и которые приходится определять эмпирически, можно узнать, лишь столкнувшись с ними в какой-то конкретной задаче. Для решения таких проблем нужна система тестов, которая позволит определять неявные характеристики конкретного ГПУ, а также позволит определить производительность ГПУ на базовых и оптимизированных версиях некоторых стандартных вычислительных и/или алгоритмических ядер и выявить возможные пути для оптимизации других ядер такого типа.

Цель настоящей статьи — создание системы тестов, позволяющих измерять различные явные и неявные характеристики графических ускорителей. Создаваемые тесты должны позволять измерять такие характеристики ГПУ, как задержка и пропускная способность различных типов памяти, эффективность атомарных операций и размер кэш-линии.

Тесты требуется апробировать в системах с ГПУ различных производителей. В качестве тестовых ГПУ в работе используются Tesla C1060, Tesla C2050, Radeon HD5830, а также ГПУ NVidia Tesla M2050, установленные в кластере “ГрафИТ!” НИВЦ МГУ [3].

[1] Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, Ленинские горы, д. 1, стр. 4, 119991, Москва; мл. науч. сотр., e-mail: adinetz@gmail.com

[2] Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, д. 1, стр. 52, 119991, Москва; студент, e-mail: shvets.pavel@gmail.com

© Научно-исследовательский вычислительный центр МГУ им. М. В. Ломоносова

В качестве языка для реализации системы тестов был выбран Nemerle и набор расширений NUDA (Nemerle Unified Device Architecture) [4], позволяющий писать программы для ГПУ с минимальными отличиями от обычных без ущерба для производительности. Система расширений NUDA в качестве нижнего уровня использует язык и среду исполнения OpenCL [5]. Таким образом, использование NUDA позволяет упростить техническую часть работы и обеспечить возможность работы системы тестов на графических ускорителях и от основных производителей (NVidia и AMD).

Уже существуют некоторые похожие системы, но они не в полной мере реализуют требуемый функционал. Например, в работе [6] описывается набор тестов, измеряющих разные неявные параметры ГПУ архитектуры GT200 от NVidia. Имеется ряд тестов, специфичных только для ГПУ от NVidia. Однако этот набор тестов не поддерживает архитектуру AMD. Набор же тестов [7] измеряет разные неявные параметры ГПУ архитектуры RV670, RV770 и RV870 от AMD, специфичных только для этих архитектур, и работает только на архитектурах AMD. Набор тестов SHOC [8] является переносимым набором бенчмарков; кроме того, для него имеются результаты измерений на большом числе карточек AMD и NVidia. В этой системе достаточный набор практических задач и бенчмарков, но недостаточно внимания уделено низкоуровневым тестам.

Ни одна из этих систем тестов не исследует все интересующие нас аспекты; например, ни один из них не исследует эффективность атомарных операций. Эти аспекты реализованы в созданной нами системе тестов. Кроме того, разрабатываемый набор будет совмещать в себе и тесты, и примеры оптимизаций, а также будет переносим между карточками различных производителей. Наконец, при помощи переносимого средства программирования могут быть созданы, в том числе, и тесты специфичных для конкретных архитектур параметров.

Тест эффективности атомарных операций. В этом тесте сравнивается эффективность локальных и глобальных атомарных операций с обычными неатомарными операциями. Локальные атомарные операции работают с локальной памятью и в связи с этим гарантируют атомарность только в пределах одного блока нитей. Глобальные атомарные операции работают с глобальной памятью и гарантируют атомарность выполнения среди всех нитей. Исследуется также влияние конфликтности шаблона доступа к памяти на эффективность.

Результаты сравнения атомарных операций с бесконфликтным доступом и атомарных операций с доступом “все-в-один” показывают отличие производительности на порядок за исключением локальных атомарных операций на карточке HD5830, которые показывают себя довольно хорошо. В дополнение к этому глобальные атомарные операции показывают в 5–10 раз меньшую эффективность, чем локальные атомарные операции.

Тест для измерения задержки глобальной, локальной и текстурной памяти. Глобальная память на ГПУ обладает очень большой латентностью, которая обычно покрывается за счет большого числа одновременно работающих нитей. Локальная память работает быстрее, но меньше по размеру. Текстурная память физически расположена там же, где глобальная память, но для нее используется другое расположение элементов массива и другой механизм кэширования. Созданный тест выводит результат и в секундах, и в пересчете на такты графического процессора для удобства оценки.

Результаты показывают, что локальная память, как и ожидалось, дает гораздо лучшие результаты, чем глобальная. Текстурная память на карточках C2050 и C1060 показывает себя хуже, чем для карточки HD5830 (задержки на чтение на порядок больше), зато C2050 выигрывает при сравнении результатов глобальной памяти с преимуществом примерно в 50%.

Тест пропускной способности глобальной, локальной и текстурной памяти. У ГПУ есть несколько важных особенностей работы с глобальной памятью.

1. Невозможно использовать всю пропускную способность памяти при малом количестве нитей.

2. Объединения запросов к памяти (coalescing). Если все нити в пределах полуварпа обращаются к одному участку памяти, то запросы объединяются в одну транзакцию, что дает выигрыш по производительности.

3. Наличие кэш-памяти. Полноценная кэш-память появилась в серии ГПУ C2050 от компании NVidia. В ряде случаев она оказывает положительное влияние на производительность.

Тесты учитывают эти особенности и исследуют зависимость пропускной способности от числа нитей для разных стандартных шаблонов обращения к памяти. В тестах присутствует специальная категория тестов “со сдвигом”: эти тесты учитывают объединение запросов к памяти, но по мере своих возможностей пытаются нивелировать воздействие кэш-памяти. Набор тестов текстурной памяти аналогичен тестам глобальной памяти за исключением технических тонкостей работы с двухмерными текстурами.

Локальная память расположена на самом чипе ГПУ, поэтому она гораздо меньше по объему, гораздо

быстрее глобальной и не кэшируется. Локальная память организована в виде банков, при доступе к которым могут возникать конфликты, снижающие пропускную способность. Тесты исследует зависимость скорости доступа к локальной памяти от конфликтности шаблона доступа и числа нитей.

Обращения к памяти в исследуемом шаблоне могут быть либо только чтениями, либо только записями, либо и чтениями, и записями. Адреса, по которым выполняется обращение, могут идти случайно, последовательно или последовательно со сдвигом, т.е. разные нити обращаются заведомо так, чтобы обращения не попали в одну кэш-линию. Таким образом, получаем девять различных шаблонов доступа в память. Псевдокод теста чтения представлен ниже.

```
int index = thread_id;
for(int i = 0; i < N; i++){
  index = data[index];
}
CountTime();
```

Результаты работы теста с использованием глобальной памяти для последовательного доступа и доступа со сдвигом показывают, что C2050 получает очень большое преимущество над остальными за счет наличия полноценного кэша (в 5 раз бóльшая пропускная способность достигается скоростью в 500 Гб/сек). В тесте, который игнорирует эффекты кэша, это преимущество исчезает.

В тесте случайного доступа карточка HD5830 показывает приблизительно на 50% бóльшую пропускную способность, чем карточка C2050. Случайные чтения: C2050 — 2.5 Гб/сек, HD5830 — 5.5 Гб/сек, C1060 — 2 Гб/сек. Случайная запись: C2050 — 9 Гб/сек, HD5830 — 15 Гб/сек, C1060 — 2.5 Гб/сек.

Тест эффективности последовательного чтения со случайной записью и случайного чтения с последовательной записью. Некоторые вычислительные алгоритмы при переносе на архитектуру ГПУ можно реализовать в двух вариантах: последовательное чтение со случайной записью или случайные чтения с последовательной записью. Был реализован имитирующий это тест, и результаты показали, что большой разницы в целом нет.

Тест для определения размера кэш-линии. Данный тест производит определенное количество чтений из глобальной памяти с разным шагом, по результатам замеров которых можно сделать предположение о размере кэш-линии на ГПУ. Псевдокод теста представлен ниже.

```
for(int step = 1; i < STEP_MAX; step++){
  int t = 0;
  for(int i = 0; i < N; i++){
    t += data[i*step];
  }
CountTime();
}
```

Результаты работы теста на ГПУ C2050 показали, что при увеличении шага чтения до какого-то момента увеличивается время; это связано с тем, что все меньше и меньше запросов попадают в одну кэш-линию. Точность не позволят со всей определенностью указать размер кэш-линии, но его размер однозначно меньше 32 слов (график после 32 линейный). Результаты работы теста на карточке C1060 показали лишь случайную структуру и небольшой разброс по времени, что позволяет предположить отсутствие кэша.

Тест когерентного доступа в память. Для традиционных систем существует тест APEX-MAP [9], позволяющий оценить эффективность работы с памятью некоторых классов задач. Была сделана попытка адаптировать этот тест под графические ускорители в последовательном и параллельном вариантах, но этот тест не дал показательных результатов и даже не смог достигнуть 20% максимальной пропускной способности. Это является еще одним доказательством того, что графические процессоры требуют другого подхода к эффективному программированию.

Недостатком традиционного теста APEX-MAP является то, что он не учитывает влияния когерентности обращений к памяти от разных нитей. В рамках системы тестов реализуется специальный тест памяти, учитывающий эту особенность. В новом тесте по памяти перемещаются группы потоков и варьируется разрозненность чтений внутри каждой группы.

Разработанный специальный тест памяти позволяет задействовать всю пропускную способность памяти и демонстрирует гораздо бóльшую зависимость от когерентности обращений групп потоков к памяти.

Тест производит измерение производительности в зависимости от размера группы совместных потоков и количества потоков из группы, которые читают одинаковые ячейки памяти (TPEIG, Threads Per

Element In Group). Например, при TPEIG = 32 и размере группы, равном 64, потоки с 64*N по 64*N+31 читают один элемент данных, а потоки с номерами от 64*N+32 до 64*N+63 — еще один, при этом читаемые элементы отстоят на расстоянии, равном TPEIG, т.е. в данном случае 32. Адреса, по которым читают элементы потоки из разных групп, вообще говоря, никак не связаны. Псевдокод теста представлен ниже.

```
int start  =  thread_n ^ (thread_n                      & (group_size-1));
int offset = (thread_n ^ (thread_n & (reads_count-1))) & (group_size-1);

for(int i = 0; i < ITERATIONS_COUNT; i++){
  start = data[start+offset];
}
CountTime();
```

На карточках C1060 десятой серии и карточке HD5830 производительность при обращении в один элемент большого числа потоков резко падает, что скорее всего связано с банками памяти. На карточках C2050 этот эффект отсутствует, предположительно из-за наличия кэш-памяти.

На всех карточках производительность растет при увеличении размера группы, пока не происходит насыщение 16/32 потока. Это соответствует архитектуре ГПУ — когерентность обращений нитей в пределах варпа сильно влияет на пропускную способность.

В качестве дополнительного исследования к данному тесту была добавлена зависимость от временно́й и пространственной локальности обращений по аналогии с оригинальным тестом APEX-MAP. Временна́я и пространственная локальность влияют на пропускную способность не так сильно, как изменение когерентности, но все-таки вносят свой вклад. Можно сделать вывод, что при оптимизации работы с памятью надо больше внимания уделять когерентности доступа, а после этого проверять локальность обращений.

Результаты по использованию некоторых из описанных тестов для определения однородности кластера "ГрафИТ!". Запуск некоторых тестов на нескольких карточках из кластера "ГрафИТ!" показал, что кластер обладает хорошей однородностью по памяти — разброс времени всех запусков для всех карточек приблизительно одинаков и составляет меньше 1%. Похожий тест для измерения однородности по вычислениям показал аналогичные результаты.

Заключение. Реализован переносимый набор тестов и проведен сбор данных с различных графических ускорителей. Получен ряд весьма интересных результатов, таких как существенное влияние кэш-памяти на пропускную способность некоторых шаблонов доступа к памяти или оценка латентности доступа к разным типам памяти. Текущая версия системы тестов свободно доступна в исходных кодах в Интернете [11].

Планами дальнейшего развития проекта является написание набора тестов-примеров оптимизаций, сравнивающих время работы неоптимизированных стандартных вычислительных ядер, таких как умножение матриц или преобразование Фурье, с версиями, оптимизированными под архитектуру графических ускорителей.

Работа выполнена при поддержке РФФИ (проект 11–07–93960-ЮАР_а) и Министерства образования и науки РФ (государственный контракт № 07.514.11.4017).

СПИСОК ЛИТЕРАТУРЫ

1. https://www-01.ibm.com/chips/techlib/techlib.nsf/products/Cell_Broadband_Engine
2. http://www.clearspeed.com/
3. http://parallel.ru/
4. http://www.khronos.org/opencl/
5. http://sourceforge.net/projects/nuda/
6. *Wong H., Papadopoulou M., Sadooghi-Alvandi M., Moshovos A.* Demystifying GPU microarchitecture through microbenchmarking // Trans. of IEEE Int. Symp. on Peformance Analysis & Software. Toronto: IEEE, 2010. 235–246.
7. *Taylor R., Li X.* A micro-benchmark suite for AMD GPUs // Proc. of the 39th Int. Conf. on Parallel Processing Workshops. San Diego: ACM Press, 2010. 387–396.
8. *Danalis A., Marin G., McCurdy C., Meredith J.S., Roth P.C., Spafford K., Tipparaju V., Vetter J.* The Scalable Heterogeneous Computing (SHOC) benchmark suite // Proc. of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units. Oak Ridge: ACM Press, 2010. 63–74.
9. http://www.prace-project.eu/documents/17_apexmap_vw.pdf
10. http://developer.amd.com/gpu/AMDAPPSDK/Pages/default.aspx
11. http://sourceforge.net/projects/gpuperfomance/

Поступила в редакцию
01.11.2011