Колпаков А.А.
Муромский институт (филиал) федерального государственного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23
E-mail: desT.087@gmail.com

Модель прогнозирования производительности гетерогенной вычислительной системы

Современные графические процессоры (graphic processor unit, GPU) – это параллельные процессоры. Точнее, они известны как потоковые процессоры, поскольку они способны выполнять различные функции в потоке входящих данных. Они представляют собой усовершенствованные архитектуры, которые предназначены для параллельной обработки данных (в первую очередь графических). На текущий момент они являются чрезвычайно мощными программируемыми процессорами, возможностями архитектуры MIMD с некоторыми ограничениями.

По мере развития технологий, языков и аппаратного обеспечения исследователи смогли использовать дополнительную гибкость графических процессоров при развертывании неграфических приложений на GPU (GPGPU), особенно при обработке изображений. Более подробно история развития GPGPU представлена в работе [1].

Дальнейшим импульсом развития стало появление CUDA, среды разработки GPGPU на основе C от NVIDIA. CUDA позволяет разработчикам, незнакомым с графическим программированием, писать код, который может быть выполнен на графическом процессоре. CUDA предоставляет необходимые абстракции для разработчика для написания многопоточных программ с небольшим знанием или без знания графических API. С тех пор для графических процессоров разработано множество реализаций распараллеленных приложений, многие из которых предлагают значительное ускорение по сравнению с последовательными реализациями на процессоре.

Модель, которая приведена в данной работе, представляет собой комбинацию известных моделей параллельных вычислений. Учитывая сложную архитектуру графического процессора, ни одна из этих моделей не является полной, и требуется комбинация из них наряду с несколькими расширениями. При разработке модели использовались:

1. Модель PRAM [2];
2. Модель BSP [3];
3. Модель QRQW [4].

Программы CUDA записываются в единицах, называемых ядрами. Нити начинаются синхронно в начале каждого ядра и синхронизируются в конце каждого ядра. Таким образом, основной единицей синхронизации в программе CUDA является ядро. Это очень близко подходит к модели параллельных вычислений BSP с неявным вызовом для синхронизации в конце каждого ядра. Однако следует обратить внимание, что в то время как в модели BSP синхронизация выполняется через регулярные интервалы в L единиц времени, представленная модель устраняет это требование. Учитывая отсутствие какой-либо инфраструктуры маршрутизации в графическом процессоре, модель BSP используется только в том, что касается понятия супершагов [3].

Окончательное уравнение выглядит следующим образом:

$$T(K) = \frac{N_B(K) \cdot N_w(K) \cdot N_t(K) \cdot C_T(K)}{N_C \cdot D \cdot R} \quad (1)$$

Поскольку каждое ядро может иметь различную структуру по числу блоков, warp-ов на каждый блок и т. д., эти величины определяются в соответствии с ядром.

Все параметры разработанной модели приведены в таблице 1.

Таблица 1. Список параметров разработанной модели

Параметр	Описание
D	Глубина конвейера ялра
N_c	Количество ядер на SM
R	Тактовая частота GPU
$C_t(K)$	Максимальное количество тактов, потребляемое любой нитью в ядре K
N_t	Количество потоков в warp = 32
N_w	Количество warp-ов на блок
$N_B(K)$	Количество блоков на ядро
K_i	i-е ядро на графическом процессоре
T(K)	Время, затраченное ядром K
T(P)	Время, затраченное программой P

В данной работе представлена модель прогнозирования производительности гетерогенной компьютерной системы в телекоммуникациях. Главное ее достоинство – это адекватная оценка возможного времени работы алгоритма при различных параметрах работы GPU, что позволяет оценить время выполнения всей задачи в целом без необходимости проведения экспериментальных исследований. Стоит заметить, что оценка, полученная с использованием разработанной модели, могла бы быть более точной, однако для этого требуется информация об аппаратной реализации механизмов работы графического процессора, которая, к сожалению, производителем не предоставляется.

Литература

1. Owens, J. D. A survey of general-purpose computation on graphics hardware. / J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. E. Lefohn, T. J. Purcell. //Computer Graphics Forum. – 2007. – vol. 26(1). – P. 80–113.

2. Fortune, S. Parallelism in Random Access Machines / S. Fortune, J. Wyllie. // Proceedings of 10th Annual ACM Symposium on Theory of Computing (STOC), ACM New York, NY, USA. – 1978. – P. 114-118.

3. Valiant, L. G. A Bridging Model for Parallel Computation / L. G. Valiant. // Communications of the ACM. – 1990. – vol. 33, no. 8. – P. 103-111.

4. Gibbons P. B. The Queue-Read Queue-Write PRAM Model: Accounting for Contention in Parallel Algorithms / P. B. Gibbons, Y. Matias, V. Ramachandran. //SIAM Journal of Computation. – 1999. – vol. 28, no. 2. – P. 733-769.

5. CUDA C Programming Guide [Электронный ресурс]. – Режим доступа: http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (20.11.2017).

6. Helman, D. R. Designing Practical Efficient Algorithms for Symmetric Multiprocessors / D. R. Helman, J. JaJa. // Lecture Notes in Computer Science 1619, International Workshop ALENEX’99. – 1999. – pp. 37-56.

7. Kolpakov, A.A. Advanced mixing audio streams for heterogeneous computer systems in telecommunications / A.A. Kolpakov, Y.A. Kropotov.//CEUR Workshop Proceedings, 2017, Vol. 1902, pp. 32-36.

8. Колпаков, А. А. Теоретическая оценка роста производительности вычислительной системы при использовании нескольких вычислительных устройств / А.А. Колпаков // В мире научных открытий. – 2012. – №1. – С. 206-209.