

В дальнейшей работе планируется создать набор функций-обработчиков, связанных с обработкой изображений, рассмотреть и учесть архитектурные особенности вычислительных систем СКИФ-МГУ «Чебышев» и «Ломоносов».

Работа ведётся при поддержке Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг.

Список литературы

1. NetCDF Tutorial. – URL: <http://www.unidata.ucar.edu/software/netcdf/docs>.
2. HDF5 Specifications. – URL: <http://www.hdfgroup.org/HDF5/doc/Specs.html>.
3. Сайт BlueGene /P, установленного в МГУ. – URL: <http://hpc.cs.msu.su>.
4. Scalable I/O and analytics / A. Choudhary [et al.] // Journal of Physics: Conference Series, Vol. 180, No. 1. doi: 10.1088/1742-6596/180/1/012048.
5. Библиотека mpix. – URL: <http://www.mpix.com>.

**¹М.В. Андреев, ¹Э.Р. Галеев, ¹А.Л. Штангеев,
¹А.В. Юлдашев, ²А.А. Яковлев**

¹Уфимский государственный авиационный технический университет
²ООО «РН-УфаНИПИнефть», г. Уфа

ОПЫТ ПОРТИРОВАНИЯ ГЕОСТАТИСТИЧЕСКОГО СИМУЛЯТОРА НА АРХИТЕКТУРУ GPU СРЕДСТВАМИ ТЕХНОЛОГИИ PGI ACCELERATOR

В последние годы геостатистическое моделирование резервуаров становится все более востребованным компонентом исследований, связанных с оптимизацией стратегии разработки нефтегазовых месторождений. Геостатистика позволяет строить детализированные количественные модели неоднородностей

строения резервуаров в тех их частях, где геофизические параметры ненаблюдаемы и их значения неизвестны.

Недавно были начаты работы по созданию высокопроизводительного симулятора для решения задач геостатистического моделирования. В основу симулятора положен новый метод построения обусловленных геостохастических геологических моделей по скважинным данным с использованием спектрального представления стационарных случайных полей [1]. В рамках проводимых работ была разработана параллельная версия симулятора для традиционных многоядерных вычислительных систем. В нашем докладе представлены результаты портирования симулятора на архитектуру GPU средствами технологии PGI Accelerator.

Одним из перспективных путей снижения времени расчетов является использование графических процессоров (GPU), в частности, производства компании NVIDIA ввиду их высокой пиковой производительности и широкой доступности. В настоящее время наиболее популярной технологией программирования вычислений общего назначения на графических процессорах производства компании NVIDIA является CUDA. Основными преимуществами технологии CUDA являются ее доступность, наличие хорошей документации, набор готовых инструментов, кроссплатформенность [2]. Однако разработка параллельных программ средствами CUDA требует от программиста детального понимания архитектуры GPU. Кроме того, при портировании на архитектуру GPU существующего программного обеспечения вычислительные ядра необходимо существенно перерабатывать.

Отмеченных недостатков лишена более высокоуровневая технология PGI Accelerator, которая поддерживается компиляторами Fortran и C99 компании Portland Group International (PGI) [3]. Данная технология позволяет программировать вычисления на GPU с поддержкой CUDA путем добавления в исходную программу директив препроцессора в стиле OpenMP. Компиляторы с поддержкой PGI Accelerator способны анализировать

структуру программы и данных, на основе указанных программистом директив распределять части программы между CPU и GPU, а также осуществлять отображение итераций циклов *for* на ядра графического процессора.

Таким образом, использование технологии PGI Accelerator не требует значительной переработки исходного кода программного обеспечения и может быть полезным при портировании существующего программного обеспечения. К недостаткам применения технологии PGI Accelerator относительно CUDA можно отнести ограниченность свободного использования компиляторов PGI.

Исходная параллельная версия геостатистического симулятора была реализована на языке C++ с использованием технологии OpenMP. Тестирование производительности OpenMP-версии на различных многоядерных системах на базе процессоров Intel и AMD показало достаточно высокую эффективность параллельной реализации: на 4 потоках ускорение расчета реальных моделей составило от 3,3 до 3,5.

Профилирование исходной версии симулятора показало, что 98 процентов времени расходуется на выполнение четырех функций: двух функций, отвечающих за построение матрицы ковариации (A и B), функции преобразования Фурье (C), а также функции стохастического моделирования (D). В целях портирования на архитектуру GPU с каждой из функций были выполнены следующие преобразования:

- функции были переписаны с языка C++ на C в связи с тем, что технология PGI Accelerator поддерживается только компиляторами языков C и Fortran;
- с помощью директивы *#pragma acc region* в функциях были выделены максимально возможные области акселерации для выполнения на GPU;
- с помощью директивы *#pragma acc data region* и ее спецификаторов были определены массивы данных, которые необходимо передавать на GPU либо получать обратно;
- с помощью директивы *#pragma acc for* и ее спецификаторов были выбраны циклы, выполнение которых переносится

на GPU, а также заданы оптимальные с точки зрения времени выполнения параметры распределения итераций циклов на ядра GPU.

Первоначально портируемые функции компилировались отдельно с помощью компилятора языка C от PGI и компоновались в библиотеку так, чтобы оставшаяся часть кода могла быть скомпилирована альтернативным компилятором языка C++. Однако, для того чтобы получить рабочую многопоточную версию программы, где каждый из потоков OpenMP работает с отдельным GPU, при сборке потребовалось использовать исключительно компиляторы PGI.

Тестирование различных версий симулятора производилось в среде ОС OpenSuSE 10.2 на двухпроцессорной рабочей станции Fujitsu-Siemens Computers CELSIUS V840 на базе двухъядерных процессоров Opteron 2214 с двухпроцессорной видеокартой GeForce GTX 295. Сборка исходной OpenMP-версии симулятора под архитектуру CPU осуществлялась компилятором Intel C++ Compiler, сборка полученной версии под архитектуру CPU+GPU – компиляторами PGI C/C++.

Результаты тестирования различных версий симулятора

Платформа	Количество потоков OpenMP	Время выполнения функций, с				
		A	B	C	D	A+B+C+D
CPU	1	235,7	426,6	167,9	384,2	1 214,4
	4	105,6	123,5	42,0	96,7	367,8
CPU+GPU	1	34,4	8,0	7,0	37,0	86,4
	2	17,3	4,1	3,7	18,7	43,7

Результаты тестирования, приведенные в таблице, показывают, что использование GPU позволяет снизить время выполнения функций в 5–30 раз. Суммарное время выполнения портированных функций снижается более чем в 8 раз. Также необходимо отметить, что использование двух процессоров видеокарты дает практически двукратный выигрыш во времени.

В итоге использование технологии PGI Accelerator позволило за достаточно короткое время получить рабочую версию симулятора для выполнения на GPU с поддержкой CUDA. Применение GPU дало существенное ускорение при решении задач геостатистического моделирования.

Список литературы

1. Байков В.А., Бакиров Н.К., Яковлев А.А. Новые подходы к теории геостатистического моделирования // Вестн. УГАТУ. – 2010. – Т. 14, № 2. – С. 209–216.
2. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. – М.: ДМК Пресс, 2010. – 232 с.
3. PGI Fortran & C Accelerator Programming Model. – URL: http://www.pgroup.com/lit/whitepapers/pgi_accel_prog_model_1.2.pdf

**Т.А. Багаутдинов¹, В.П. Гергель¹, А.В. Горшков¹,
И.И. Фикс^{1,2}, М.Ю. Кириллин^{1,2}**

¹Нижегородский государственный университет им. Н.И. Лобачевского

²Институт прикладной физики РАН, г. Нижний Новгород

МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ СВЕТА В МНОГОСЛОЙНОЙ СРЕДЕ МЕТОДОМ МОНТЕ-КАРЛО

В настоящее время лазерные методы получили широкое распространение в бесконтактной неразрушающей диагностике внутренней структуры различных оптически неоднородных объектов, в частности, они находят применение в медицине, биофизике, науках о материалах, физике атмосферы и других областях.

Для повышения эффективности современных методов лазерной диагностики, а также для разработки новых методов необходимо подробное изучение особенностей процесса распространения света в различных средах, включая биоткани.