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: kaf-eivt@yandex.ru

Разработка структур систем обработки данных для гетерогенных компьютерных систем с повышенной производительностью

GPGPU-программу можно условно представить при помощи следующих множеств:

−множество шейдеров, выполняющих вычисления;

−множество переменных, управляющих вычислениями;

−множество данных, над которыми проходят вычисления и в которые заносятся их результаты;

−множество инструкций, которые запускают тот или иной шейдер, предоставляют ему на вход те или иные данные и выводят результат в некоторую текстуру.

Как известно, элементарным примитивом для визуализации, с которым работает графический процессор, является треугольник. При этом с каждой вершиной треугольника можно связать ограниченный набор произвольных данных, например ее цвет, нормаль и другие пользовательские данные. С самим примитивом может быть ассоциировано до 8 текстур – одно-, двух-, и трехмерные изображения. Структурная схема обработки данных на основе вершинных и пиксельных программ представлена на рисунке 1.

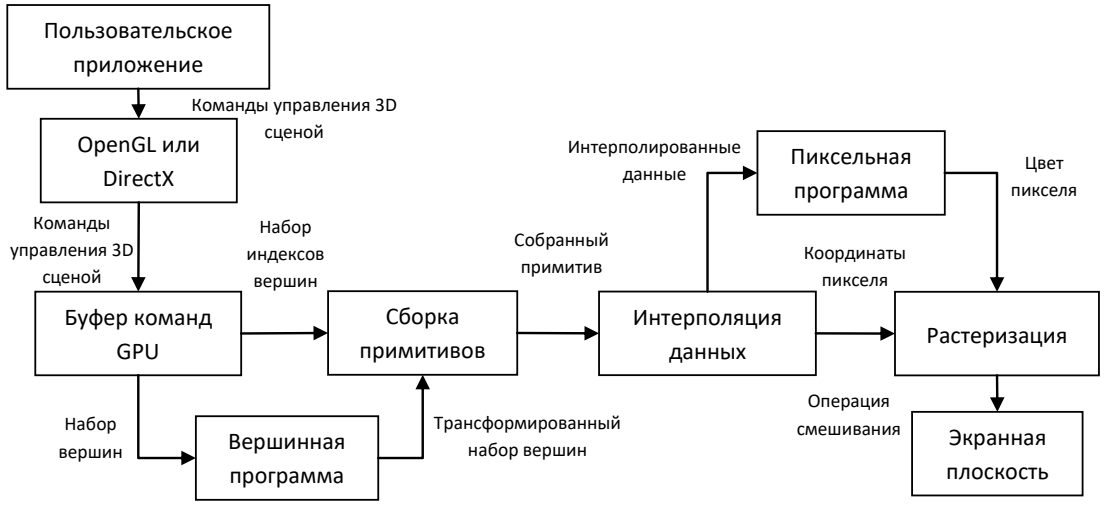


Рисунок 1. Схема разработанной локальной вычислительной сети в в *симуляторе сети передачи данных Packet Tracer*

В ходе исследования эффективности разработанных алгоритмов обработки информации была использована задача нахождения нулевых битовых векторов, которая решается с применением генетических алгоритмов [42]. При решении указанной задачи основное время работы занимают параллельные вычисления значений функций приспособленности различных особей, операций скрещивания и мутации. Используемый алгоритм ее решения имеет свойства, характерные для многих генетических алгоритмов:

1. Представление особи в виде битовой строки.
2. Малое число логических операций при вычислении функции приспособленности, выполнении мутации и скрещивания.
3. Последовательный доступ к памяти.

Данные свойства позволяют эффективно использовать вычисления на графическом процессоре.

Операция мутации стандартна для таких особей — изменение значения одного случайного бита. В качестве операции скрещивания используется одноточечный кроссовер.

Функцией приспособленности особи является число единиц в ней. Соответственно, необходимо вывести идеальную особь с нулевой функцией приспособленности. Существует алгоритм, позволяющий вычислять число единиц в 32-битном числе, используя только арифметические операции.

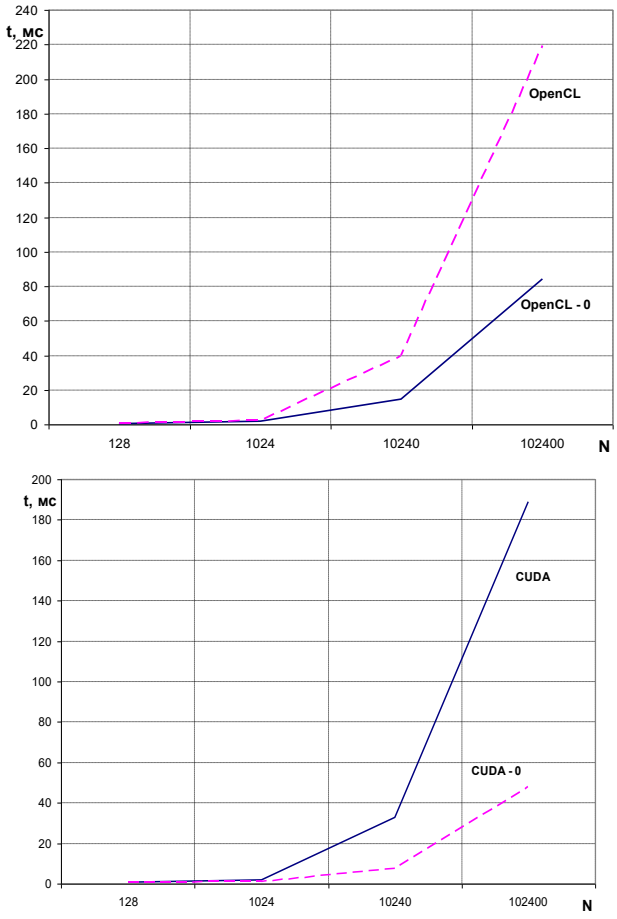
Для проведения экспериментальной оценки эффективности работы алгоритма повышения производительности обработки данных использовалась тестовая компьютерная система следующей конфигурации: центральный процессор Intel Core 2 Quad Q9400 (2.66GHz), ОЗУ 8GB, графическая карта Nvidia GeForce GTX560 2Gb 336 потоков, операционная система Windows 7 x64, компилятор MS Visual Studio 2008 в release режиме.

Исследовалось среднее время, потраченное на получение нового поколения для различных размеров задачи и числа особей в поколении.

Для этого запускалось несколько итераций получения очередного поколения (около 100 – 1000 запусков) и общее время, потраченное на всю работу алгоритма, делилось на число полученных поколений.

При исследовании производительности первой тестовой задачей изменялось количество 32-битных целых чисел в массиве (M) и число параллельных потоков (N).

Исследовалось среднее время t, потраченное на получение нового поколения для различного количества 32-битных целых чисел в массиве и числа параллельных потоков. Исследования проводились с использованием технологий OpenCL и NVIDIA CUDA [4]. Результаты исследований среднего времени, потраченного на получение нового поколения, для параметра M = 10 приведены на рисунке 2.



а б

Рисунок 2. Среднее время, потраченное распараллеленной системой на получение нового поколения, при M =10, а: OpenCL – базовый алгоритм, OpenCL-O – разработанный алгоритм, б: CUDA– базовый алгоритм, CUDA-O – разработанный алгоритм.

На рис. 10 графики OpenCL и CUDA показывают время выполнения базового алгоритма, OpenCL-O и CUDA-O – с применением разработанного алгоритма повышения производительности обработки данных. Как видно из графиков, при значении параметра M = 10 применение разработанного алгоритма оптимизации дает рост производительности: в случае применения OpenCL время обработки для 128 потоков сокращается с 1,08 мс до 0,75 мс и с 219 мс до 84,2 мс для 102400 потоков. В случае применения NVIDIA CUDA время обработки сокращается с 0,85 мс до 0,74 мс для 128 потоков и со 189 мс до 48,4 мс для 102400 потоков [4].

Список используемых источников:

1. Современные проблемы вычислительной математики и математического моделирования. Т. 1: Вычислительная математика. / Под ред. Бахвалов Н. С., Воеводин В. В. – М.: Наука, 2005. – 342 с.

2. Колпаков, А. А. Аспекты оценки увеличения производительности вычислений при распараллеливании процессоров вычислительных систем / А.А. Колпаков // Методы и устройства передачи и обработки информации. – 2011. – №1(13). – С. 124-127.

3. 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.

4. Кропотов, Ю. А. Экспериментальные исследования закона распределения вероятности амплитуд сигналов систем передачи речевой информации / Ю.А. Кропотов // Проектирование и технология электронных средств. – 2006. – Т.4. – С. 37-42.

5. Кропотов, Ю. А., Проскуряков А. Ю., Белов А. А., Колпаков А. А. Модели, алгоритмы системы автоматизированного мониторинга и управления экологической безопасности промышленных производств / Ю.А. Кропотов, А. Ю. Проскуряков, А. А. Белов, А.А. Колпаков // Системы управления, связи и безопасности. – 2015. – №2. – С. 184-197.

6. Кропотов, Ю. А., Белов А. А., Проскуряков А. Ю., Колпаков А. А. Методы проектирования телекоммуникационных информационно–управляющих систем аудиообмена в сложной помеховой обстановке / Ю.А. Кропотов, А. А. Белов, А. Ю. Проскуряков, А.А. Колпаков // Системы управления, связи и безопасности. – 2015. – №2. – С. 165-183.

7. Kropotov, Ju.A. Identification of Models for Discrete Linear Systems with Variable, Slowly Varying Parameters / V.A. Ermolaev, V.T. Eremenko , O.E. Karasev , Ju.A. Kropotov //Journal of Communications Technology and Electronics. – 2010. – vol. 55, № 1. – P. 52-57.