

**Негосударственное аккредитованное некоммерческое
Частное образовательное учреждение высшего образования
«АКАДЕМИЯ МАРКЕТИНГА И СОЦИАЛЬНО-ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ – ИМСИТ»
(г. Краснодар)**

**Факультет информатики и вычислительной техники
Кафедра математики и вычислительной техники**



УТВЕРЖДАЮ
Председатель НМС,
проректор по учебной работе,
профессор

Н.Н. Павелко

16 апреля 2018г.

**Б1.В.ДВ.10.02
ПРОГРАММИРОВАНИЕ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ**

рабочая программа учебной дисциплины для
студентов направления подготовки 09.03.04

Программная инженерия

Направленность (профиль) программы: «Информационно-вычислительные
системы»

Квалификация (степень выпускника) бакалавр

**г. Краснодар
2018**

Рабочая программа составлена с учётом Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия (уровень бакалавриата), утверждённого приказом Министерства образования и науки Российской Федерации от 12.03.2015 г. № 229

Составитель _____  В.В. Бужан

Согласовано:

Проректор по качеству, доцент



К.В. Писаренко

Рецензенты:

Левченко В.И., к.т.н., доцент, доцент кафедры автоматизации производственных процессов КубГТУ

Суриков А.И., директор ООО «1С-КОНСОЛЬ»

Рабочая программа рассмотрена на заседании кафедры математики и вычислительной техники от 19.03.2018 г., протокол №8

Зав. кафедрой математики и вычислительной техники, к.т.н., доцент _____  Н.С.Нестерова

Рабочая программа утверждена на заседании Научно-методического совета Академии от 16.04.2018 г., протокол №8.

1. Цели освоения дисциплины

В курсе рассматриваются такие вопросы как архитектура графических процессоров, ее отличие от архитектуры центральных процессоров общего назначения, классы задач, которые возможно решать с использованием гибридных вычислительных систем и пр.

Студенты на практике получают опыт программирования графических процессоров, достаточный для эффективного решения практических задач по моделированию в различных областях науки и техники в рамках своей магистерской диссертации.

Дисциплина «Программирование графических процессоров» имеет своей целью научить студентов:

- реализовывать эффективные программы для выполнения на графических процессорах;
- использовать средства разработки программ для графических процессоров;
- использовать графические процессоры из программ на различных языках программирования;
- оценивать имеющиеся алгоритмы с целью их адаптации под графические процессоры;
- применять эффективные подходы по адаптации готовых программ под графические процессоры.

Курс также направлен на формирование общекультурных и профессиональных компетенций в соответствии с ФГОС ВПО по данному направлению подготовки.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование графических процессоров» относится к дисциплинам по выбору (вариативная часть, дисциплина по выбору). Результаты освоения дисциплины могут быть использованы студентом при подготовке своей магистерской диссертации.

Требования к слушателям:

- знание языка Си;
- умение вести разработку программ или в MS Visual Studio, или из командной строки UNIX Shell;
- умение пользоваться табличным процессором, программой GNUplot, или другим средством визуализации данных.

Дисциплины, последующие по учебному плану:

- Итоговая государственная аттестация.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины «Программирование графических процессоров» у обучающегося формируются компетенции:

ПК – 3 владением навыками использования различных технологий разработки программного обеспечения

В результате освоения дисциплины обучающийся должен:

Знать:

- Архитектуру графических процессоров и ее отличие от архитектуры центрального процессора общего назначения.
- Основной состав и функциональность оптимизированных под GPU библиотек.
- Основные понятия, связанные с технологией CUDA.

Уметь:

- Оценивать алгоритм на возможность его адаптации под гибридные вычислительные системы.
- Выполнить реализацию заданного алгоритма для выполнения на графическом процессоре.
- Переносить существующие программные коды для их эффективного выполнения на графических процессорах.
- Использовать оптимизированные библиотечные функции.

Владеть:

- Основными принципами оптимизации программ под графические процессоры.
- Способами отладки и профилирование программ под графические процессоры.
- Инструментальными средствами разработки, отладки и профилирования приложений под графические процессоры.

4. Структура и содержание дисциплины «Программирование графических процессоров»

Общая трудоемкость дисциплины составляет 2 зачетных единиц, 72 учебных часа, из них 33 аудиторных часа.

№ п/п	Раздел дисциплины	Семестр	Планируемые	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
1	Введение	1	1	1	1		2	
2	Программный интерфейс CUDA	1	2	1	1	4	6	Прием задач.
3	Программирование на уровне драйвера	1	3	1	1	1	3	Прием задач.
4	Работа с текстурной, разделяемой, константной памятью	1	4	1	1	4	6	Прием задач.
5	Оптимизация программы для графического процессора	1	5-6	2	2	4	8	Прием задач.
6	Использование нескольких графических процессоров	1	7-8	2	2	4	8	Прием задач. Оценка за контрольную неделю.
7	Адаптация программы для выполнения на графическом процессоре	1	9	1	1	3	5	Прием задач.
8	Оптимизированные математические библиотеки	1	10	1	1	4	6	Прием задач.
9	Введение в OpenACC	1	13-14	2	2	3	7	Прием задач. Оценка за контрольную неделю.
10	Использование графических процессоров	1	13-14	2	2	2	6	Прием задач.

	для математических расчетов из Fortran, Java, C#							
11	Введение в OpenCL	1	15	1	1	2	4	Прием задач.
12	Применение графических процессоров для решения прикладных задач	1	16	1	1	5	7	Крайний срок сдачи всех задач (задания)
13	Промежуточная аттестация	1	17		1 КСР	3	4	Дифференцированный зачет
	Итого:			16	16 + 1КСР	39	72	

5. Образовательные технологии

Учебный курс «Программирование графических процессоров» предусматривает применение различных методов обучения, сочетая вместе с традиционными методами обучения (лекции, практические занятия) дистанционные технологий обучения (вебинары).

На лекциях студентам преподается материал по текущей теме. В случае необходимости часть материала преподается в виде разбора конкретных ситуаций на примерах применения читаемого материала к решению практических задач, например, учет иерархии памяти графического процессора при решении задачи перемножения матриц. Каждая лекция начинается и заканчивается ответами преподавателя на вопросы студентов по изученному ранее и на текущей лекции материалу. К чтению лекций могут привлекаться представители ведущих российских и зарубежных компаний. Лекционный материал курса представлен в виде мультимедиа презентаций. Презентации всех лекций, ссылки на дополнительную литературу, вебинары выкладываются для студентов в информационной системе НГУ «студент-преподаватель» - MyNSU (my.nsu.ru).

На практических занятиях студентам предлагается реализовать программный код - выполнить задание, состоящие из 6 практических задач (см.раздел 6.1). На первом практическом занятии озвучивается формулировка задания и критерии его оценки. В течение семестра вводятся три контрольных срока сдачи задач. Два контрольных срока совпадают с контрольными неделями в университете. Третий – на 16 неделе обучения. В процессе приема решения задачи студенту задаются вопросы на понимание работы программы и связанного с ней лекционного материала. В случае некорректной работы программы или непонимания того, как она работает,

сдаваемая задача отправляться на доработку. Результат усвоения материала (количество сданных задач) публикуется в системе MyNSU.

В целях контроля степени усвоения курса в течение семестра, в соответствии с учебными планами, в Университете устанавливаются контрольные недели. Оценки по дисциплине за контрольные недели определяются количеством правильно сданных задач (см. раздел 6).

Самостоятельная работа студента в течение семестра предполагает выполнение, доработку практических задач, и изучение дополнительного материала, необходимого для выполнения Задания (см. разделы 7б. и 7в.).

В соответствии с учебным планом по дисциплине предусмотрена промежуточная аттестация в форме дифференцированного зачета. Оценка за курс выставляется по результатам теста на 17 неделе обучения и количеству успешно сданных задач. Вопросы к тесту формируются на основе лекционного и дополнительно изученного студентом материалов. Тест состоит как из вопросов с кратким ответом, так и из открытых вопросов, требующих развернутый ответ. Примеры открытых вопросов см. в разделе 6.2.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В соответствии с учебными планами, устанавливаются формы контроля: текущий контроль студентов в течение семестра и промежуточная аттестация в виде дифференцированного зачета в конце семестра.

Текущий контроль включает в себя – прием задач в течение семестра и выставление оценок за контрольные недели, с учетом правильно сданных задач в контрольные сроки.

Критерии степени усвоения материала студентом:

Оценка за первую контрольную неделю:

- «отлично» - принято три задачи и более;
- «хорошо» - принято две задачи;
- «удовлетворительно» - принято одна задача;
- «неудовлетворительно» - не принято ни одной задачи.

Оценка за вторую контрольную неделю:

- «отлично» - принято пять задач и более;
- «хорошо» - принято четыре задачи;
- «удовлетворительно» - принято две задачи;
- «неудовлетворительно» - принято менее двух задач.

Оценка за 16-ю неделю обучения:

- «отлично» - принято шесть задач;
- «хорошо» - принято пять задач;
- «удовлетворительно» - принято три задачи;
- «неудовлетворительно» - принято менее трех задач.

Студенты, получившие оценку «неудовлетворительно» за 16-ю неделю обучения, к сдаче теста не допускаются.

Оценка за курс (диф.зачет) выставляется по результатам теста и количества успешно сданных задач, как среднеарифметическое значение оценок за 16-ю неделю обучения и теста.

Критерии степени усвоения материала студентом по результатам теста:

- «отлично» - более 90% правильных ответов;
- «хорошо» - от 80% до 90% правильных ответов;
- «удовлетворительно» - от 60% до 80% правильных ответов;
- «неудовлетворительно» - менее 60% правильных ответов или несданный тест.

6.1. Задачи для практических занятий:

1. Выделить на GPU массив `arr` из 10^9 элементов типа `float` и инициализировать его с помощью ядра следующим образом: `arr[i] = sin((i%360)*Pi/180.0)`. Скопировать массив в память центрального процессора и посчитать ошибку `err = sum_i(abs(sin((i%360)*Pi/180.0) - arr[i]))/109`. Провести исследование зависимости результата от использования функций: `sin`, `sinf`, `__sinf` и ключа компиляции

-use-fast-math. Объяснить результат. Проверить результат при использовании массива типа double.

2. Реализовать программу для накладывания фильтров на изображения. Возможные фильтры: размытие, выделение границ, избавление от шума. Реализовать три варианта программы, а именно: на глобальной памяти, с применением разделяемой памяти и текстур. Сравнить время выполнения.
3. Модифицировать предыдущую программу так, чтобы использовались все имеющиеся в распоряжение программы GPU. Программа должна определять количество доступных графических процессоров и распределять работу по ним.
4. При помощи метода наименьших квадратов найти окружность в изображении. Для каждой случайной выборки точек организовать их обработку на GPU. Случайные выборки организовать при помощи библиотеки CURAND.
5. Ray Tracing. Реализовать генерацию сцены, состоящей минимум из двух сфер и минимум одной плоскости. На выбор реализовать или преломление или отражение луча от сфер. На плоскости должна быть натянута текстура. Минимальный размер результирующего изображения 640x480 точек.
6. Для заданной программы на Фортране выполнить перенос части кода на графический процессор. Возможны два варианта: используя возможности компилятора PGI Accelerated Fortran или с помощью вызова внешней функции с ядром на Си\C++.

6.2. Примеры открытых вопросы для теста:

1. Назовите основные отличия между GPU и CPU.
2. Чем отличается архитектура Tesla и Fermi. Особенности программирования.
3. Что такое CUDA?
4. Что такое варп, потоковый блок?
5. Какие типы памяти есть на графическом процессоре? Перечислите их характеристики.

6. Что такое текстура? Текстура и тип double.
7. Что такое «вычислительная совместимость» (Compute compatibility)?
8. Оптимизированные библиотеки CUDA.
9. Что такое конфликты по банкам памяти и методы борьбы с ними.
10. Что такое «спиллинг регистров»?
11. Потоки. Совмещение вычислений с передачей данных.
12. Программирование гибридных систем с несколькими GPU.
13. Средства профилирования приложений на CUDA.
14. Оптимизация CUDA приложений. Подходы.
15. Программирование графических процессоров на Java, Fortran, C#.
16. Реализация операции редукции на графическом процессоре.
17. Реализация перемножения матриц на графическом процессоре.
18. Реализация функции размытия изображения на графическом процессоре.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Павловская Т.А. С/C++. Программирование на языке высокого уровня: Учебник для вузов – СПб.: Питер, 2014. – 464 с.
<http://www.studfiles.ru/preview/397655/>
2. Бужан В.В. Программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 230100.62 – Информатика и вычислительная техника. – Краснодар: ИМСИТ, 2013. – 84 с.
3. Программные и аппаратные средства информатики/Царев Р.Ю., Прокопенко А.В., Князьков А.Н. - Краснояр.: СФУ, 2015. - 160 с.
<http://znanium.com/catalog.php?bookinfo=550017>
4. Гавриков М.М. Теоретические основы разработки и реализации языков программирования: учеб. Пособие/ под ред. А.Н. Иванченко.- М.: КНОРУС, 2016.-184 с.

б) дополнительная литература:

1. Р. Гонсалес, Р. Вудс, «Цифровая обработка изображений», Техносфера, 2006, ISBN: 5-94836-028-8
2. Боресков А.В., Харламов А.А., «Основы работы с технологией CUDA», ДМК Пресс, 2010, ISBN: 978-5-94074-578-5

в) Интернет-ресурсы:

1. «NVidia CUDA Programmers guide»
[<http://developer.nvidia.com/cuda-downloads>]
2. «Истории успеха» [http://www.nvidia.ru/page/case_studies.html]
3. «High Performance Computing - Supercomputing with Tesla GPUs»,
[http://www.nvidia.com/object/tesla_computing_solutions.html]
4. «OpenCL - The open standard for parallel programming of heterogeneous systems» [<http://www.khronos.org/opencv/>]
5. «Компьютерная обработка изображений»
[http://aco.ifmo.ru/el_books/image_processing/]
6. Вебинары по технологии CUDA
[<http://www.gotdotnet.ru/blogs/parallel-computing/>]

8. Материально-техническое обеспечение дисциплины

- Наличие в компьютерах или на выделенном сервере устройств, поддерживающих технологию CUDA (<http://developer.nvidia.com/cuda-gpus>);
- Наличие установленных последних стабильных версий драйверов и библиотек (<http://developer.nvidia.com/cuda-downloads>)
- Наличие среды разработки под Си\C++ и Фортран;
- Наличие PGI Accelerated Fortran (опционально);
- Наличие средств для визуализации результата (табличный процессор или gnu plot).
- Наличие доступа в Интернет с рабочих мест.