Федеральное государственное бюджетное образовательное
учреждение высшего образования
Московский государственный университет имени М.В.Ломоносова
Филиал Московского государственного университета имени М.В.Ломоносова
в городе Сарове

УТВЕРЖДАЮ
Директор филиала МГУ в
городе Сарове

_____/В.В.
Воеводин/

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Наименование дисциплины (модуля):

«Высокопроизводительные вычисления на кластерах с использованием графических ускорителей NVIDIA»

Уровень высшего образования:
Подготовка магистров (неинтегрированная магистратура)

Направление подготовки (специальность):

«Прикладная математика и информатика» (01.04.02)

Направленность (профиль) ОПОП:

«Суперкомпьютерные технологии математического моделирования и обработки данных»

Форма обучения:

Очная

Саров 2021

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.04.02 "Прикладная математика и информатика" программы магистратуры в редакции приказа МГУ от 30 декабря 2020 г. №1366

1. Место дисциплины (модуля) в структуре ОПОП ВО:

Дисциплина «Высокопроизводительные вычисления на кластерах с использованием графических ускорителей NVIDIA» входит в базовую часть магистерской программы «Суперкомпьютерные технологии математического моделирования и обработки данных». Данная дисциплина относится к блоку «Программное обеспечение современных вычислительных комплексов».

2. Входные требования для освоения дисциплины (модуля), предварительные условия:
Учащиеся должны владеть знаниями по курсам, связанным с основами программирования и системами программирования.

3. Результаты обучения по дисциплине (модулю):

Формируемые компетенции	**Планируемые результаты обучения**
Способен совершенствовать и реализовывать новые математические и компьютерные методы решения прикладных задач (ОПК-2). Способен разрабатывать и реализовывать проекты, предусматривая и учитывая проблемные ситуации и риски на всех этапах выполнения проекта (УК-3). Способен комбинировать и адаптировать современные информационно-коммуникационные технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности (ОПК-4)	Знать: основные особенности программно-аппаратной архитектуры CUDA, основные понятия и современные тенденции в мире современных графических вычислений, а также основные аппаратные особенности и характеристики современных графических ускорителей компании NVIDIA. Уметь: реализовывать базовые программы для GPU при помощи программной модели CUDA и OpenACC, производить профилировку и анализ эффективности GPU приложений, оценивать производительность и потенциал оптимизации различных классов программ для графических ускорителей NVIDIA. Владеть: базовыми навыками программирования графических ускорителей NVIDIA с использованием программных моделей CUDA и OpenACC, базовыми навыками использования основных библиотек, входящих в состав CUDA Toolkit, базовыми понятиями, навыками профилирования программ, написанных для GPU. Иметь навык (опыт): опыт работы с современными графическими

	ускорителями NVIDIA архитектур Pascal, Volta и Ampere, навыки работы на современных кластерах и суперкомпьютерах, построенных на основе графических ускорители NVIDIA, навыки разработки и оптимизации различных прикладных CUDA приложений.

4. Объем дисциплины (модуля).

Объем дисциплины составляет 4 зачетные единицы, всего 144 часа.

72 часа составляет контактная работа с преподавателем – 36 часов занятий лекционного типа, 36 часов научно-практических занятий.

72 часа составляет самостоятельная работа учащегося.

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий:

В курсе дается обзор основных понятий, методов, концепциях и проблемах связанных ускорением решаемых задач с использованием графических процессоров (GPU) NVIDIA. Изучаются основы программирования с использованием технологии CUDA, приводятся сведения о типах памяти GPU, рассматриваются библиотеки и инструменты, входящие в комплект разработчика CUDA Toolkit, вопросы отладки, профилирования и оптимизации CUDA-программ. Также рассматривается стандарт параллельного программирования OpenACC и особенности программирования многопроцессорных систем с несколькими GPU.

5.1. Структура дисциплины (модуля) по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий (в строгом соответствии с учебным планом)

Наименование разделов и тем дисциплины (модуля), **Форма промежуточной аттестации**	**Номинальные трудозатраты обучающегося**		**Всего академических часов**	**Форма текущего контроля успеваемости*** ***(наименование)***
	Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, академические часы	**Самостоятельная работа обучающегося, академиче**		

по дисциплине (модулю)	Занятия лекционного типа	Занятия семинарскго типа	ские часы		
Тема 1. Введение в программирование GPU NVIDIA.	3	1	5	9	*опрос*
Тема 2. Программная модель CUDA.	3	3	8	14	*опрос, выполнение практического задания*
Тема 3. Иерархия памяти CUDA.	3	4	8	15	*опрос, выполнение практического задания*
Тема 4. Работа с разделяемой памятью CUDA.	3	5	9	17	*опрос, выполнение практического задания*
Тема 5. Интегрированные среды разработки с поддержкой CUDA, инструменты для отладки и профилирования.	3	4	8	15	*опрос, выполнение практического задания*

Тема 6. Дополнительные возможности работы с памятью.	3	2	5	10	*опрос, выполнение практического задания*
Тема 7. CUDA Streams.	3	3	8	14	*опрос, выполнение практического задания*
Тема 8. Программирование систем с несколькими графическими процессорами (Multi-GPU).	3	3	8	14	*опрос, выполнение практического задания*
Тема 9. Библиотеки с поддержкой CUDA.	2	3	8	13	*опрос, выполнение практического задания*
Тема 10. Основы программирования GPU NVIDIA с использованием стандарта OpenACC.	3	2	5	10	*опрос, выполнение практического задания*

Тема 11. Реализация графовых задач на графических ускорителях.	2	6	5	13	*опрос, выполнение практического задания*
Итоговая индивидуальная работа по теме «реализация графового алгоритма поиска в ширину на графических ускорителях NVIDIA»	—	—	30	30	—
Промежуточная аттестация (зачет(ы) и (или) экзамен(ы))	5	—	—	—	— —
Итог	**36**	**36**	**72**	**144**	—

5.2. Содержание разделов (тем) дисциплины

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин

1.	**Тема 1. Введение в программирование GPU NVIDIA.**	Базовые понятия программно-аппаратной архитектуры CUDA. Основные различия центральных процессоров и графических ускорителей. Основы работы с суперкомпьютером (заход, компиляция, запуск задач), использующим графические ускорители.
2.	**Тема 2. Программная модель CUDA.**	Основные понятия параллельной архитектуры CUDA: нить, warp, блок, грид, ядро. Использование CUDA API для взаимодействия между хостом и устройством. Передача данных между хостом и устройством. Параллельное сложение векторов на GPU средствами CUDA.
3.	**Тема 3. Иерархия памяти CUDA.**	Иерархия памяти современных графических ускорителей. Основы работы с глобальной памятью. Шаблоны доступа к глобальной памяти. Механизм транзакций и работа с кэшами L1 и L2 графических ускорителей.

4.	**Тема 4. Работа с разделяемой памятью CUDA.**	Различные способы и шаблоны работы с разделяемой памятью. Синхронизация нитей блока при использовании разделяемой памяти. Использование разделяемой памяти для ускорения решения прикладных задач.
5.	**Тема 5. Интегрированные среды разработки с поддержкой CUDA, инструменты для отладки и профилирования.**	Использование средства профилировки nvprof для анализа эффективности CUDA приложений. Примеры использования Nvidia Visual profiler для различных реальных приложений. Дистанционный сбор данных профилировки с удаленного сервера. Основные подходы к анализу эффективности и оптимизации GPU программ на основе рассмотренных средств.
6.	**Тема 6. Дополнительные возможности работы с памятью.**	Pageble и Page-locked память. Unified память. Pinned-память. Использование атомарных операций в CUDA-ядрах.
7.	**Тема 7. CUDA Streams.**	Использование CUDA Streams для реализации асинхронных операций: асинхронных пересылок, запусков ядер в конкурентном режиме.

8.	**Тема 8. Программирование систем с несколькими графическими процессорами (Multi-GPU).**	Использования модели OpenMP и механизма переключения контекстов для создания программ, использующих несколько графических ускорителей. Разработка программ на основе MPI для кластеров, построенных на основе графических ускорителей.
9.	**Тема 9. Библиотеки с поддержкой CUDA.**	Обзор интерфейсов и основных возможностей прикладных библиотек CUBLAS, cuSPARSE, cuRAND, nvGRAPH, thrust и др. Пример использования данных библиотек для реализации матричного умножения.
10.	**Тема 10. Основы программирования GPU NVIDIA с использованием стандарта OpenACC.**	Основные директивы OpenACC, принципы копирований данных. Примеры адаптации программ с помощью OpenACC.
11.	**Тема 11. Реализация графовых задач на графических ускорителях.**	Примеры графовых задач и алгоритмов их решения. Типовые шаблоны доступа к данным в графовых задачах. Проблема балансировки параллельной нагрузки для графов реального мира. Основные подходы к оптимизации графовых задач.

6. Фонд оценочных средств (ФОС, оценочные и методические материалы) для оценивания результатов обучения по дисциплине (модулю).

6.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости, критерии и шкалы оценивания (в отсутствие утвержденных соответствующих локальных нормативных актов на факультете)

Темы практических заданий.

1) Реализация базового сложения векторов с использованием CUDA.
2) Реализация базового транспонирования плотных матриц: сравнение различных подходов к выбору размеров конфигураций ядра и сетки.
3) Оптимизация ядра графового алгоритма поиска кратчайших путей от заданной вершины-источника.
4) Реализация блочного транспонирования плотных матриц с использование разделяемой памяти.
5) Реализация умножения плотных матриц с использованием глобальной (6 вариантов расстановки циклов) и разделяемой памяти. Дополнительная оптимизация алгоритма матричного умножения.
6) Реализация задач сложения векторов и умножения плотных матриц с использованием Pinned-памяти и Unified-памяти.
7) Реализация задачи сложения векторов с использованием технологии CUDA-потоков.
8) Реализация бечмарка random memory access и его модификация с использованием технологии CUDA-потоков.
9) Реализация задачи сложения векторов с использованием Multi-GPU.
10) Реализация задачи умножения матриц при помощи библиотек CUBLAS, CURAND, Thrust.
11) Реализация задачи поиска кратчайших путей в графе при помощи библиотеки NVGRAPH.
12) Умножение плотных матриц и сложение плотных векторов с использованием технологий CUDA и OpenACC.

Критерии оценивания каждого из практических заданий:
1) задание сдано преподавателю в обозначенный в курсе лекций срок, проведено детальное исследование эффективности и производительности полученной реализации — отл.
2) задание сдано преподавателю в обозначенный в курсе лекций срок, но не приведено достаточного обоснования производительности и эффективности разработанной реализации — хор.
3) задание сдано преподавателю позже указанного в курсе срока, либо при его решении используются заведомо неоптимальные подходы к реализации, обсуждаемые в рамках курса — удовл.
4) задание не сдано до момента зачета/экзамена по курсу — неудовл.

6.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации по дисциплине (модулю), критерии и шкалы оценивания (в отсутствие утвержденных соответствующих локальных нормативных актов на факультете)

Практические задания:

1) Реализация трехмерного stencil-ядер с использованием разделяемой памяти, Multi-GPU, unified(или pinned) памяти.
2) Конкурсная реализация задачи поиска в ширину в графе.

Критерии оценивания каждого из практических заданий:
1) задание сдано преподавателю в обозначенный в курсе лекций срок, проведено детальное исследование эффективности и производительности полученной реализации — отл.
2) задание сдано преподавателю в обозначенный в курсе лекций срок, но не приведено достаточного обоснования производительности и эффективности разработанной реализации — хор.
3) задание сдано преподавателю позже указанного в курсе срока, либо при его решении используются заведомо неоптимальные подходы к реализации, обсуждаемые в рамках курса — удовл.
4) задание не сдано до момента зачета/экзамена по курсу — неудовл.

Промежуточные тестирования:
1) Тестирование по базовым возможностям работы с графическими ускорителями: темы 1 — 5 представленной программы, 20 теоретических вопросов.
2) Тестирование по дополнительным возможностям программирования графических ускорителей: темы 6 — 10 представленной программы, 20 теоретических вопросов.

Критерии оценивания каждого из практических заданий:
1) 18 и более правильных ответов — отл.
2) 15 и более правильных ответов — хор.
3) 12 и более правильных ответов — удовл.
4) менее 12 правильных ответов — неудовл.

---

7. Ресурсное обеспечение:
7.1. Перечень основной и дополнительной литературы

Основная литература:

1) Попова Н.Н., Колганов А.С., Снытников А.В., Математическое моделирование и программная модель CUDA, издательство ООО "МАКС Пресс" (Москва) , ISBN ISBN 978-5-317-05911-8, 176 с., 2018 г.
2) Боресков А. В. и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учебное пособие. Издательство Московского университета, 2012
3) Сандерс Дж., Кэндрот Эд. Технология CUDA в примерах. Введение в программирование графических процессоров. ДМК Пресс, 2011 г.

Дополнительная литература:
1) Pharr M (Ed.) GPU Gems 2. Programming Techniques for. High-Performance Graphics and. General-Purpose Computation. Addison-Wesley, 2005.

7.2. Перечень лицензионного программного обеспечения, в том числе отечественного производства (подлежит обновлению при необходимости)

1) Программное обеспечение для подготовки слайдов лекций MS PowerPoint
2) Программное обеспечение для создания и просмотра pdf-документов Adobe Reader
3) Пакет putty (или openssh)
4) CUDA Programming Toolkit со следующими компонентами: компилятор NVCC, NVIDIA command line profiler, NVIDIA Visual profiler, cuRAND, cuBLAS, cuSparse, NVGRAPH, thrust libraries, CUDA samples.

7.3. Перечень профессиональных баз данных и информационных справочных систем (подлежит обновлению при необходимости)

Нет.

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1) CUDA Programming tookit: https://developer.nvidia.com/cuda-toolkit
2) CUDA Developer additional resources: https://developer.nvidia.com/additional-resources
3) Stackoverflow, CUDA questions: https://stackoverflow.com/questions/tagged/cuda

7.5. Описание материально-технического обеспечения.

Медиапроектор и экран для проведения лекций-презентаций. Суперкомпьютерный комплекс на основе графических ускорителей для выполнения практических заданий.

8. Соответствие результатов обучения по данному элементу ОПОП результатам освоения ОПОП указано в Общей характеристике ОПОП.

9. Разработчик (разработчики) программы.
Аспирант Афанасьев Илья Викторович (afanasiev_ilya@icloud.com)