


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**

**Наименование дисциплины:**

**Параллельные вычисления**

---

**Уровень высшего образования:**

**магистратура**

---

**Направление подготовки / специальность:**

**02.04.02 "Фундаментальная информатика и информационные технологии" (3++)**

---

**Направленность (профиль)/специализация ОПОП:**

**Суперкомпьютерные технологии и фундаментальная информатика**

---

**Форма обучения:**

**очная**

---

Саров 2022

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 02.04.02 "Фундаментальная информатика и информационные технологии" программы магистратуры в редакции приказа МГУ от 30 декабря 2020 г. №1366



1. Дисциплина относится к вариативной части магистерской программы ОПОП ВО, изучается в третьем семестре.

2. Входные требования для освоения дисциплины (модуля), предварительные условия (если есть):

Учащиеся должны владеть знаниями по операционным системам, компьютерным сетям, базам данных, дискретной математике и основам кибернетики.

3. Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.

<b>Содержание и код компетенции</b>	<b>Индикатор (показатель) достижения компетенции</b>	<b>Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций</b>
Способность разрабатывать и реализовывать проекты, предусматривая и учитывая проблемные ситуации и риски на всех этапах выполнения проекта (УК-3).	<b>УК-3</b> Разрабатывает и реализовывает проекты, предусматривая и учитывая проблемные ситуации и риски на всех этапах выполнения проекта (УК-3).	УК-3- 31 Знать: современные проблемы математики, вычислительной математики;  УК-3- У1 Уметь: формализовывать теоретическую проблему, нахождения способа и алгоритма её решения; УК-3-В1 Владеть: научной картиной мира;
Способность проводить теоретическое исследование и экспериментальный анализ эффективности функционирования и методов организации вычислений для многопроцессорных вычислительных систем, проводить оценку масштабируемости параллельных программ (МПК-4).	<b>МПК-4</b> Проводит теоретическое исследование и экспериментальный анализ эффективности функционирования и методов организации вычислений для многопроцессорных вычислительных систем, проводить оценку масштабируемости параллельных программ	МПК-4- 31 Знать: взаимосвязь и фундаментальное единство естественных наук. МПК-4_У1 Уметь: эффективно использовать на практике теоретические компоненты науки: понятия, суждения, умозаключения, законы; МПК-4-В1 Владеть: математическим моделированием физических задач;
Способность комбинировать и адаптировать современные информационно-коммуникационные	<b>ОПК-4</b> Проводит теоретическое исследование и экспериментальный анализ эффективности функционирования и методов	ОПК-4-31 Знать: постановку проблем физико-математического моделирования; ОПК-4-У1 Уметь: абстрагироваться от

технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности (ОПК-4)	организации вычислений для многопроцессорных вычислительных систем, проводить оценку масштабируемости параллельных программ	несущественных влияний при моделировании реальных физических ситуаций; планировать оптимальное проведение расчета. ОПК-4-В1 Владеть: навыками самостоятельной работы на современных вычислительных комплексах.
--	---	---

4. Формат обучения лекционные занятия проводятся с использованием меловой доски и проектора.

5. Объем дисциплины (модуля) составляет 4 з.е., всего 144 академических часов, в том числе 72 академических часов, отведенных на контактную работу обучающихся с преподавателем, 72 академических часов на самостоятельную работу обучающихся.

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

Наименование и краткое содержание разделов и тем дисциплины (модуля),  Форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе	
		Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, часы	Самостоятельная работа обучающегося, часы <i>(виды самостоятельной работы – эссе, реферат, контрольная работа и пр. – указываются при необходимости)</i>

		Занятия лекционного типа*	Занятия семинарского типа*	Всего	
<b>Тема 1. Современные суперкомпьютерные архитектуры</b> Виды параллелизма - MIMD, SIMD, потоковая обработка. Гибридная кластерная система и ее характеристики. Виды вычислительных устройств и систем. Топологии вычислительных систем. Уровни параллелизма и средства разработки. Параллелизм CPU ядра. Архитектура вычислительного узла. Принципы работы оперативной памяти. Свойства каналов передачи данных. Оценка времени передачи данных между вычислительными узлами. Области применения многопроцессорных систем. Суперкомпьютерное моделирование сплошной среды. Примеры газодинамических расчетов.	15	8	-	8	7
<b>Тема 2. Параллельные алгоритмы, основные понятия и принципы построения</b> Внутренний параллелизм. Оценка сложности, ускорения, эффективности, масштабируемости параллельного алгоритма. Оценка времени выполнения программы. Закон Амдала. Сверхлинейное ускорение. Статическая и динамическая балансировка загрузки процессоров. Методы: сдвигание, геометрический параллелизм, конвейерный параллелизм, коллективное решение, диффузная балансировка. Статическая и динамическая балансировка загрузки при решении задач, использующих нерегулярные расчетные сетки.	15	8	-	8	7
<b>Тема 3. Технологии распараллеливания для MIMD систем с общей и распределенной памятью, организация взаимодействия параллельных процессов</b> Виды обменов данными (MPI). Организация обмена данными, оптимизация. Скрытие обменов за вычислениями. Процессы и	15	8	-	8	7

<p>потоки в операционных системах. Потоки и семафоры стандарта Posix. Критические секции и блокировка потоков. Синхронизация процессов. Основные директивы OpenMP, простейшие примеры. Накладные расходы и проблемы. Способы устранения конфликтов по доступу к данным на примере сеточного метода.</p>					
<p><b>Тема 4. Гетерогенные вычисления на процессорах и ускорителях</b>  Устройство графического процессора GPU. Поточковая обработка. Вычислительный стандарт OpenCL. Организация доступа к памяти. Примеры реализации вычислений. Методика портирования kernel-кода. Многоуровневая декомпозиция. Сокращение обменов за вычислениями на ускорителях. Примеры реализации вычислений.</p>	15	8	-	8	7
<p><b>Тема 5. Декомпозиция сеточных графов</b>  Примеры расчетных сеток. Критерии декомпозиции графов. Критерий 1: классический критерий декомпозиции графа. Критерий 2: выделение обособленных доменов. Критерий 3: минимизация максимальной степени домена. Критерий 4: обеспечение связности графов каждого из доменов. Декомпозиция на основе исходной нумерации узлов. Рекурсивная бисекция. Декомпозиция регулярных графов. Методы декомпозиции произвольных графов. Иерархическая декомпозиция. Координатная бисекция. Спектральная бисекция. Алгоритм инкрементного роста доменов. Декомпозиция больших сеток. Двухуровневая стратегия обработки и хранения сеток. Пакеты декомпозиции графов.</p>	18	10	-	10	8
<p><b>Тема 6. Обработка и визуализация данных, заданных на решетках</b>  Клиент-серверная технология. Online или Offline визуализация: плюсы и минусы. Этапы визуализации. Визуализация изоповерхностей. Аппроксимация изоповерхности. Способы описания триангуляции. Метод редукции. Параллельные алгоритмы построения аппроксимирующих триангуляций. Многоуровневое огрубление больших сеток. Примеры визуализации. Ввод-вывод сеточных данных. Соотношение времени чтения данных и времени их обработки. Распределенный ввод-вывод. Огрубление и сжатие скалярных сеточных функций: без потери точности, с контролируемой потерей точности. Форматы хранения данных. Пакеты визуализации данных большого объема.</p>	18	10	-	10	8
<p><b>Тема 7. Отказоустойчивые алгоритмы для высокопроизводительных систем</b>  Контрольные точки. Локальные контрольные точки. Принцип локальности для алгоритмов на основе явных разностных схем.</p>	18	10	-	10	8

Функции MPI, обеспечивающие возможность работы в условиях отказа части вычислительных узлов. Эффективность отказоустойчивых алгоритмов. Распределенное хранение локальных контрольных точек. Совмещение во времени этапов распределенной записи локальных контрольных точек и вычислений.					
<b>Тема 8. Параллельные алгоритмы решения прикладных задач</b> Параллельные алгоритмы сортировки данных. Алгоритмы генерации псевдослучайных чисел для суперкомпьютерных приложений. Балансировка загрузки процессоров при решении задач численного интегрирования, многомерной оптимизации, решении систем линейных алгоритмов. Динамическая балансировка загрузки процессоров при моделировании процессов горения.	18	10	-	10	8
<b>Промежуточная аттестация</b> – результаты выполнение практических заданий по созданию модельных параллельных приложений и исследованию их параллельной эффективности на кластерной системе.					10
<b>Промежуточная аттестация – устный экзамен</b>					2
<b>Итого</b>	<i>144</i>	<i>72</i>			<i>72</i>

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине (модулю)

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

Примеры заданий для самостоятельной работы студентов

Практическое самостоятельное задание № 1

Многопоточная реализация солвера CG для СЛАУ с разреженной матрицей, заданной в формате ELLPACK.

Требования к отчету:

1. Титульный лист, содержащий

1.1 Название курса

1.2 Название задания

1.3 Фамилию, Имя, Отчество(при наличии)

1.4 Номер группы

1.5 Дата подачи

Содержание отчета:

2 Описание задания и программной реализации

2.1 Краткое описание задания

2.2 Краткое описание программной реализации как организованы данные, какие функции реализованы (название, аргументы, назначение)

2.2.1 Указать как программа запускается (ставится в очередь) с какими параметрами, с описанием этих параметров

2.3 Описание опробованных способов оптимизации последовательных вычислений (по желанию)

3 Исследование производительности

3.1 Характеристики вычислительной системы:

описание одной или нескольких систем, на которых выполнено исследование (подойдет любой многоядерный процессор).

тип процессора, количество ядер, пиковая производительность, пиковая пропускная способность памяти. по желанию - промерять и на своем десктопе/ноуте, и на кластере

3.1.1 Указать здесь и или в следующих пунктах как программа компилировалась (каким компилятором, с какими параметрами)

3.2 Результаты измерений производительности

3.2.1 Последовательная производительность

Для каждой из трех базовых операций и для всего алгоритма солвера исследовать зависимость достигаемой производительности от размера системы  $N$ , построить графики GFLOPS от  $N$ . Несколько  $N$  достаточно:  $N=1000, 10000, 100000, 1000000$

Для повышения точности измерений, замеры времени лучше производить, выполняя набор операций многократно в цикле, чтобы осреднить время измерений. Суммарное время измерений чтобы получалось порядка нескольких секунд. Оценить выигрыш от примененной оптимизации (по желанию)

3.2.2. Параллельное ускорение

Измерить OpenMP ускорение для различных  $N$  для каждой из 3-х базовых операций и для всего алгоритма солвера. При фиксированном числе  $N$  варьируется число нитей и измеряется параллельное ускорение.

4. Анализ полученных результатов



- 4.1 Процент от пика оценить для каждой из трех базовых операций, какой процент от пиковой производительности устройства составляет максимальная достигаемая в тесте производительность
  - 4.2 Процент от достижимой производительности аналогично оценить для каждой операции процент от максимально достижимой производительности с учетом пропускной способности памяти.
- Приложение 1: исходный текст программы в отдельном c/c++ файле
- Требования к программе:
- 1 Программа должна использовать OpenMP или posix threads для многопоточного распараллеливания
  - 2 Солвер должен корректно работать, т.е. показывать быструю сходимость.

### Практическое самостоятельное задание № 2

Распределенная реализация солвера CG для СЛАУ с разреженной матрицей, заданной в формате ELLPACK

Требования к отчету:

Титульный лист, содержащий

- 1.1 Название курса
- 1.2 Название задания
- 1.3 Фамилию, Имя, Отчество(при наличии)
- 1.4 Номер группы
- 1.5 Дата подачи

Содержание отчета:

2. Краткое описание задания и программной реализации

2.1 Краткое описание задания

2.2 Краткое описание программной реализации как организованы данные, какие функции реализованы (название, аргументы, назначение)

Просьба указывать, как программа запускается с какими параметрами, с описанием этих параметров.

3. Исследование производительности

3.1 Характеристики вычислительной системы: описание одной или нескольких систем, на которых выполнено исследование.

Использование кластера в этом задании обязательно.

Просьба указывать здесь или в следующих пунктах, как программа компилировалась (каким компилятором, с какими параметрами).

3.2 Результаты измерений производительности

3.2.1 Сравнение MPI с OpenMP на многоядерном процессоре.

Для каждой из трех базовых операций и для всего алгоритма солвера сравнить ускорения на разном числе ядер, полученные в MPI и OpenMP режиме, оценить параллельную эффективность. Достаточно одного размера системы,  $N=10^6$ . Данные представить в виде таблицы.

3.2.2. Параллельное ускорение Измерить MPI ускорение для различных  $N$  порядка  $10^5, 10^6, \dots$  для каждой из 3-х базовых операций и для всего алгоритма солвера: при фиксированном числе  $N$  варьируется число процессов и измеряется параллельное ускорение. Построить графики

ускорения.

3.2.3. Масштабирование Измерить масштабирование для различных фиксированных N/P порядка  $10^4$ ,  $10^5$ ,  $10^6$ . Здесь N/P – количество неизвестных на процесс. В этом тесте число N варьируется пропорционально числу процессов, P.

Данные представить в виде таблицы и графика.

Приложение1: исходный текст программы в отдельном c/c++ файле

Требования к программе:

- 1 Программа должна использовать MPI для распараллеливания с распределенной памятью, OpenMP или posix threads для многопоточного распараллеливания (которое уже имеется из 1 -го задания)
- 2 Солвер должен корректно работать, т.е. показывать быструю сходимость.
- 3 Распараллеливание должно быть корректно.

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

#### **Вопросы к экзамену**

1. Параллелизм процессорного ядра и основные характеристики. Конвейер, out-of-order, суперскалярность и другие.
2. Основные характеристики суперкомпьютера - производительность пиковая, теоретически достижимая, пропускные способности.
3. Основные характеристики параллельного алгоритма – оценка сложности, вычислительная интенсивность, ускорение, эффективность, масштабируемость.
4. Устройство оперативной памяти, ключевые характеристики, снижение потерь и оптимизация доступа (DDR, тайминги, иерархия памяти и т. д.)
5. Многопроцессорные системы с общей памятью (UMA, NUMA, ccNUMA), устройство многопроцессорного узла, снижение потерь на NUMA эффекты.
6. Многопоточное распараллеливание Posix, OpenMP, MIMD, общие принципы и проблемы (race condition, false sharing, NUMA, affinity), способы устранения зависимости по данным в сеточных методах.
7. Многопроцессорные системы с распределенной памятью. Устройство и основные характеристики (производительность, латентность и пропускная способность сети, оценка времени передачи данных, топология системы и отображение группы процессов на ресурсы).
8. Распараллеливание с распределенной памятью, MIMD, MPI, основные принципы, виды обмена данными (p2p, групповые обмены различных видов), синхронные и асинхронные обмены
9. Геометрический параллелизм, рациональная декомпозиция расчетной сетки, организация обмена данными. Способы снижения накладных расходов (DMA, overlap, двухуровневая декомпозиция).
10. Устройство графического процессора, общие принципы, архитектура GPU, парадигма потоковой обработки, основные отличия от CPU.
11. Потоковое распараллеливание, OpenCL или CUDA, общие принципы и подходы к реализации (иерархия памяти, буфера, kernel, очереди заданий, накладные расходы)

12. Гетерогенные вычисления. Общие принципы, основные проблемы, способы реализации, многоуровневая декомпозиция.
13. Форматы хранения разреженных матриц (CSR, ELLPACK, ...), особенности операции SpMV и ее распараллеливания, способы повышения производительности (RCM reordering, coalescing, и др.)
14. Параллельное ускорение, параллельная эффективность, масштабирование, закон Амдаля, причины сверхлинейного ускорения
15. Барьер. Схемы реализации и оценки времени выполнения. Реализация барьера на основе синхронных обменов. Реализация барьера на основе семафоров Дейкстры.
16. Методы статической и динамической балансировки загрузки процессоров.
17. Метод сдваивания.
18. Метод коллективного решения.
19. Метод конвейерного параллелизма.
20. Метод диффузной балансировки загрузки.
21. Метод серверного параллелизма.
22. Отладка параллельных приложений, выполнение которых сопровождается недетерминированным потоком сообщений.
23. Параллельные алгоритмы сортировки данных. Слияние методом сдваивания, двустороннее слияние. Сети сортировки. Параллельные алгоритмы сортировки для гибридных вычислительных систем.
24. Параллельные алгоритмы генерации псевдослучайных чисел. Требования к генераторам псевдослучайных чисел. Линейно-конгруэнтные генераторы. М-последовательности. Достоинства и недостатки генераторов случайных и псевдослучайных чисел.
25. Иерархические алгоритмы разбиения графов. Локальное уточнение.
26. Метод спектральной бисекции графов.
27. Инкрементный алгоритм декомпозиции графов.
28. Адаптивное интегрирование. Метод локального стека. Метод глобального стека.
29. Отказоустойчивые алгоритмы для многопроцессорных вычислительных систем. Локальные контрольные точки. Локальное исправление ошибок при решении гиперболических систем уравнений.

### Примеры задач

1. Напишите алгоритм для вычислительной системы с распределенной памятью, определяющий сумму элементов массива. Считайте, что массив уже распределен между  $m$ -процессами. Оцените ускорение и эффективность алгоритма в предположении, что время суммирования двух элементов равно  $TauC$ , а время пересылки одного числа между процессорами равно  $TauS$ .
2. Напишите алгоритм для вычислительной системы с общей памятью, определяющий сумму элементов массива. Оцените ускорение и эффективность алгоритма в предположении, что время суммирования двух элементов равно  $TauC$ , а минимальное время обращения к семафору равно  $TauS$ .
3. Нарисуйте сеть четно-нечетного слияния Бетчера для двух упорядоченных групп элементов, содержащих 4 и 7 элементов. Чему равно число тактов работы построенной сети?
4. Определить период генератора  $u[i+1] = (5 * u[i] + 3) \bmod 1024$ .

5. Определить значение элемента номер 51 последовательности трёхразрядных двоичных чисел, формируемых генератором  $x^i \bmod (x^4 + x^3 + 1)$ .
6. Дано:  $u(t=0; i=0 \dots n)=0$ ,  $u(t, i=0)=0$ ;  $u(t=0, \dots, m; i=n)=1$ ,  $u(t+1; i=1, \dots, n-1)=f(u(t, i-1), u(t, i), u(t, i+1))$ . Напишите параллельную программу для вычислительной системы с распределенной памятью, определяющую значения  $u(t=m; i=0, \dots, n)$ . Приведите оценку ускорения и эффективности программы в предположении, что время однократного вычисления функции  $f$  равно  $\tau_C$ , а время пересылки одного числа между процессорами равно  $\tau_S$ . Укажите, какой вид параллелизма использован.
7. Напишите алгоритм барьера на основе синхронных обменов.

<b>ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)</b>				
Оценка	2	3	4	5
РО и соответствующие виды оценочных средств				
<b>Знания</b> (виды оценочных средств: устные и письменные опросы и контрольные работы, тесты, и т.п.)	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
<b>Умения</b> (виды оценочных средств: практические контрольные задания, написание и защита рефератов на заданную тему и т.п.)	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности непринципиального характера)	Успешное и систематическое умение
<b>Навыки (владения, опыт деятельности)</b> (виды оценочных средств: выполнение и защита курсовой работы, отчет по	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

## 8. Ресурсное обеспечение:

Основная учебно-методическая литература и перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Якововский М.В. Введение в параллельные методы решения задач: Учебное пособие / Предисл.: В. А. Садовничий. – М.: Издательство Московского университета, 2012. – 328 с., илл. – (Серия «Суперкомпьютерное образование»), ISBN 978-5-211-06382-2 URL: <http://lira.imamod.ru/ITTPMOPS/>
2. Лацис А.О. Параллельная обработка данных. – М.: Академия, 2010 г. 336 стр. ISBN 978-5-7695-5951-8
3. Дейкстра Э. Взаимодействие последовательных процессов. <http://khpi-iip.mipk.kharkiv.edu/library/extent/dijkstra/ewd123/index.html>
4. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002. URL: <http://a1308.ru/books/section/179>
5. Якововский М.В., Кулькова Е.Ю. Решение задач на многопроцессорных вычислительных системах с разделяемой памятью. - М.: СТАНКИН, 2004. – 30 с. [http://www.imamod.ru/~serge/arc/stud/Jacob\\_2.pdf](http://www.imamod.ru/~serge/arc/stud/Jacob_2.pdf)
6. Дональд Э. Кнут Искусство программирования. Том 3. Сортировка и поиск
7. В. Е. Карпов, А. И. Лобанов, Численные методы, алгоритмы и программы. Введение в распараллеливание [Текст] : учебное пособие для студентов высших учебных заведений, обучающихся по направлению подготовки "Прикладные математика и физика" /. - Москва : Физматкнига, 2014. - 190 с. : ил.; 21 см. - (Серия Суперкомпьютерное образование : СКО).; ISBN 978-5-89155-234-0
8. Головизнин В.М., Зайцев М.А., Карабасов С.А., Короткин И.А. Новые алгоритмы вычислительной гидродинамики для многопроцессорных вычислительных комплексов: Монография / Предисл.: В. А. Садовничий. - М.: Издательство Московского университета. 2013. - 472 с., илл.; 8 с. (цв. вклейка). - (Серия «Суперкомпьютерное образование») ISBN 978-5-211-06426-3
9. Гергель В.П., Сысоев А.В. и др., Высокопроизводительные параллельные вычисления. 100 заданий для расширенного лабораторного практикума, Издательство: Физматлит, ISBN: 5922118161, ISBN-13(EAN): 9785922118163, 248 с., 2018
10. Стивенс У. UNIX: взаимодействие процессов: СПб.: Питер; 2002, 576с. Ил.
11. Optimizing software in C++. An optimization guide for Windows, Linux and Mac platforms By Agner Fog. Technical University of Denmark. Copyright © 2004 -2018. Last updated 2018-08-18. [https://www.agner.org/optimize/optimizing\\_cpp.pdf](https://www.agner.org/optimize/optimizing_cpp.pdf)
12. MPI: A Message-Passing Interface Standard Version 3.0. 2012. <https://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf>
13. The Argonne Training Program on Extreme-Scale Computing (ATPESC) 2018 <https://extremecomputingtraining.anl.gov/archive/>
14. Pavan Balaji, et. al. Advanced MPI Programming. Tutorial at SC17? 2017. <https://www.mcs.anl.gov/~thakur/sc17-mpi-tutorial/slides.pdf>
15. OpenMP tutorial. Blaise Barney, Lawrence Livermore National Laboratory. <https://computing.llnl.gov/tutorials/openMP/>
16. Спецификации стандарта OpenMP. <https://www.openmp.org/specifications/>

17. Tim Mattson, Larry Meadows. A “Hands-on” Introduction to OpenMP. <https://www.openmp.org/wp-content/uploads/omp-hands-on-SC08.pdf>
18. Спецификации стандарта OpenCL <https://www.khronos.org/opencl/>
19. Демонстрационные примеры на OpenCL NVIDIA, <https://developer.nvidia.com/opencl>
20. OpenCL Best Practice Guide. NVIDIA [https://www.cs.cmu.edu/afs/cs/academic/class/15668-s11/www/cuda-doc/OpenCL\\_Best\\_Practices\\_Guide.pdf](https://www.cs.cmu.edu/afs/cs/academic/class/15668-s11/www/cuda-doc/OpenCL_Best_Practices_Guide.pdf)
21. AMD Accelerated Parallel Processing OpenCL. Optimization Guide. 2014. [https://developer.amd.com/wordpress/media/2013/12/AMD\\_OpenCL\\_Programming\\_Optimization\\_Guide.pdf](https://developer.amd.com/wordpress/media/2013/12/AMD_OpenCL_Programming_Optimization_Guide.pdf)
22. Горобец А. В. Параллельная технология численного моделирования задач газовой динамики алгоритмами повышенной точности, Журнал вычислительной математики и математической физики, т. 55, №4, с. 641-652, 2015.

Дополнительная учебно-методическая литература и перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Лацис А.О. Как построить и использовать суперкомпьютер. - Бестселлер, 2003 г. 274 стр. ISBN 5-98158-003-8
2. Тель Ж. Введение в распределенные алгоритмы. пер. с англ. В. А. Захарова. - М. : МЦНМО, 2009. - 616 с. : ил.
3. Эндрюс Г.Р. Основы многопоточного параллельного и распределенного программирования. — М.: Издательский дом «Вильямс», 2003. — 512 с.: ил. — Парал. тит. англ.
4. Ким А. К., Перекатов В. И., Ермаков С. Г. Микропроцессоры и вычислительные комплексы семейства «Эльбрус». — СПб.: Питер, 2013. - 272 с, ил. — ISBN 978-5-459-01697-0.
5. Михайлов Г.А., Войтишек А.В. Численное статистическое моделирование. Методы Монте-Карло Учебное пособие для студентов вузов. — М.: Академия, 2006. — 368 с. — (Университетский учебник : Прикладная математика и информатика).
6. Зорин А.В., Федоткин М.А., Методы Монте-Карло для параллельных вычислений. Учебное пособие, - М.: Издательство Московского университета, 192 с., 2013 ISBN 978-5-211-06530-7
7. A.Gorobets, S.Soukov, P.Bogdanov. Multilevel parallelization for simulating turbulent flows on most kinds of hybrid supercomputers. Computers and Fluids. Volume 173, Pages 171-177. 2018. <https://doi.org/10.1016/j.compfluid.2018.03.011>
8. Стронгин, Р.Г. Параллельные вычисления в задачах глобальной оптимизации / Р.Г. Стронгин Р.Г., В.П. Гергель, В.А. Гришагин, К.А. Баркалов. — М.: Изда-тельство Московского университета, 2013. — 280 с.
9. X.Alvarez, A.Gorobets, F.X.Trias, R.Borrell, and G.Oyarzun. HPC2 - a fully portable algebra-dominant framework for heterogeneous computing. Application to CFD. Computers and Fluids. Volume 173. Pages 285-292. 2018. <https://doi.org/10.1016/j.compfluid.2018.01.034>
10. Суков С. А., Горобец А. В., Богданов П. Б. Переносимое решение для моделирования сжимаемых течений на всех существующих гибридных суперкомпьютерах, Математическое моделирование, 2017 год, том 29, номер 8, стр. 3-16
11. Горобец А. В. Методика выполнения крупномасштабных расчетов задач газовой динамики, Математическое моделирование (РИНЦ), 2016, том 28, номер 4, стр. 77-91.

12. С. А. Суков, А. В. Горобец, П. Б. Богданов, Адаптация и оптимизация базовых операций газодинамического алгоритма на неструктурированных сетках для расчетов на массивно-параллельных ускорителях, Журнал вычислительной математики и математической физики, 2013, том 53, № 8, с. 1360–1373.

При реализации дисциплины может быть использовано следующее программное обеспечение:

1. Программный продукт Red Hat Enterprise Linux Server for HPC Compute Node for Power, LE, Self-support
2. Программный продукт Red Hat Enterprise Linux Server for HPC Head Node for Power, LE, Standard
3. Операционная система SUSE Linux Enterprise Server 11 SP4 for x86\_64
4. Операционная система Red Hat Enterprise Linux Server 5.0 for x86\_64
5. Операционная система SUSE Linux Enterprise Server 10 SP3 for ppc64
6. Операционная система Ubuntu 18.04.
7. Программное обеспечение для виртуализации Oracle VM VirtualBox
8. Операционная система ALTLinuxMATEStarterkit 9 лицензияGPL
9. Программный продукт JetBrains IntelliJ IDEA Community Edition Free Educational Licenses
10. Программный продукт JetBrains PyCharm Community Edition Free Educational Licenses
11. Программный продукт JetBrains CLion Community Edition Free Educational Licenses
12. Программный продукт UPPAAL (<http://www.uppaal.org/>) академическая лицензия
13. Программный продукт Java 8 (64-bit) Oracle Corporation
14. Программный продукт Java SE Development Kit 8(64-bit) Oracle Corporation
15. Программный продукт NetBeans IDE 8.2 NetBeans.org
16. Программный продукт Dev-C++ Bloodshed Software
17. Программный продукт CodeBlocks The Code::Blocks Team
18. Программный продукт Free Pascal 3.0.0 Free Pascal Team
19. Программный продукт Python 3.5.1 (64-bit) Python Software Foundation
20. Программный продукт R for Windows 3.2.2 R Core Team
21. Программный продукт Haskell Platform 7.10.3 Haskell.org
22. Операционная система Microsoft Windows 7 корпоративная академическая лицензия
23. Операционная система Microsoft Windows 10 Education академическая лицензия
24. Программный продукт Microsoft Project Professional 2013 академическая лицензия
25. Программный продукт Microsoft Visio Professional 2013 академическая лицензия
26. Программный продукт Microsoft Visual Studio Professional 2013 - RUS [Русский (Россия)] академическая лицензия

Информационные технологии, используемые в процессе обучения: в процессе обучения используются приложения Google Docs, среда разработки MS Visual Studio, технологии параллельного программирования MPI, OpenMP, OpenACC, DVM.

Материально-техническая база: для преподавания дисциплины аудитория, оборудованная проектором.

9. Язык преподавания.

русский

10. Преподаватель (преподаватели).

Чл.-корр.РАН, профессор Якобовский М.В., д.ф.-м.н. Горобец А.В., ассистент Жуков К.А.

11. Автор (авторы) программы.

Чл.-корр.РАН, профессор Якобовский М.В., д.ф.-м.н. Горобец А.В.