Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 23.10.2023 16:02:46
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета информационных технологий

/Д. Г. Демидов/

«28» 04 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Высокопроизводительные вычисления»

Направление подготовки
09.03.03 Прикладная информатика
Образовательная программа (профиль подготовки)
«Большие и открытые данные»

Квалификация (степень) выпускника
бакалавр

Форма обучения
очная

Москва 2022

Программа дисциплины «Высокопроизводительные вычисления» составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению **09.03.03 «Прикладная информатика»** и профилю подготовки **«Большие и открытые данные».**

Программу составил



_____/В.Г. Евтихов/

Программа дисциплины утверждена на заседании кафедры «Прикладная информатика»
«____» августа 2022 г. протокол № _____

Заведующий кафедрой
доцент, к.э.н.  /С.В.Суворов/

Программа согласована с руководителем образовательной программы по направлению подготовки **09.03.03 «Прикладная информатика»** по профилю подготовки **«Большие и открытые данные».**



_____ /С.В.Суворов/
«____» августа 2022 г.

Программа утверждена на заседании учебно-методической комиссии факультета Информационных технологий

Председатель комиссии  / Д. Г. Демидов/

«____»_____2022 г. Протокол:

Цели освоения дисциплины

Целями дисциплины являются получение представления о современных подходах, используемых при создании высоко-производительных программно-аппаратных комплексов, освоение современных методов высокопроизводительных вычислений, и, в частности, параллельного программирования.

Основное внимание уделено существующим и перспективным архитектурам высокопроизводительных вычислительных систем и передовым программным технологиям, обеспечивающим высокую производительность создаваемых программ.

Задачами дисциплины являются:

- Получение навыков, необходимых для построения высокопроизводительных параллельных программ с использованием различных технологий.
- Ознакомление с новыми перспективными технологиями разработки высокопроизводительных программ.
- Освоение навыков разработки высокопроизводительных информационных подсистем.

1. Место дисциплины в структуре ООП бакалавриата

Входными требованиями, необходимыми для освоения дисциплины, является наличие у обучающихся компетенций, сформированных при изучении дисциплины «Теоретические основы информатики», «Программирование».
Дисциплина «Высокопроизводительные вычисления» относится к дисциплинам Вариативной части (дисциплины по выбору студента) основной образовательной программы бакалавриата.

2. Перечень планируемых результатов обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций:

Код компетенции	Наименование компетенции	Индикаторы планируемых результатов обучения по дисциплине
Категория (группа) «Профессиональные компетенции и индикаторы»		
ПК-1	Способен разрабатывать и адаптировать прикладное программное обеспечение	**ПК-1.1. Знать:** требования к программному коду; языки программирования, определения и манипулирования данными. **ПК-1.2. Уметь:** составлять программный код с использованием языков программирования, определения и манипулирования данными; производить проверку и отладку программного кода; работать с системой контроля версий; разрабатывать и документировать программный интерфейс; разрабатывать технические спецификации на программные компоненты и их взаимодействие; разрабатывать тестовые наборы данных и процедур проверки работоспособности программного обеспечения; осуществлять интеграцию программных модулей и компонент и верификации выпусков программного продукта: составлять формализованные описания решений, поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных. **ПК-1.3. Владеть:** навыками оптимизации программного кода; навыками проверки работоспособности программного обеспечения; навыками анализа возможностей реализации требований к программному обеспечению; навыками проектирования программного обеспечения; навыками исправления дефектов, зафиксированных в базе данных.

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единицы, т.е. 216 академических часа (из них 180 часов – самостоятельная работа студентов).

Все они осваиваются на **четвертом** курсе бакалавриата в 8 семестре.

Лекции – 2 часа в неделю (18 часов), лабораторные работы – 4 часа в неделю (54 часов), форма контроля – экзамен.

Содержание разделов дисциплины
Содержание и темы лекций
представлены в следующей таблице.

Л-1	Основные понятия и законы высокопроизводительных вычислений	2 ак. часа
Краткое содержание (перечень рассматриваемых вопросов) лекции: • Термин производительность • Закон Амдала • Закон Густавсона – Барсиса • Закон Мура		
1. Где используются и применяются методы высокопроизводительной обработки данных? 2. Какой показатель является существенным для вычислительных систем? 3. В чем заключается закон Амдала? 4. В чем заключается закон Густавсона-Барсиса? 5. Раскройте содержание закона Мура и проанализируйте его.		
Л-2	Стек вычислительных технологий	2 ак. часа
Краткое содержание (перечень рассматриваемых вопросов) лекции: • Рассматриваются различные виды вычислений: граничные, туманные, облачные. • Распределенные и параллельные вычисления. • Виртуальные машины и виртуальные приватные сети. • Рассматриваются средства удаленного администрирования. • Изучаются архитектуры параллельных вычислений и графическая визуализация параллелизма.		
Контрольные вопросы: 1. На чем основывается стек вычислительных технологий? 2. Что понимается под распределенными вычислениями? 3. Что понимается под распределенными вычислениями? 4. Что представляет собой виртуальная вычислительная машина? 5. Охарактеризуйте распределенные вычисления. 6. Что такое суперскалярность ядра вычислительной системы? 7. В каком виде может быть представлена визуализация параллелизма? 8. Приведите краткую характеристику платформы Apache Hadoop. 9. В чем заключаются особенности файловой системы HDFS? 10. Опишите алгоритмы технологий обработки данных MapReduce. 11. Известно, что в файле конфигурации Hadoop задан размер блока 64 МБ. Файл bigdata.gz занимает 1 ГБ места в HDFS. Данный файл был выбран для некой MapReduce операции. Сколько мапперов будет создано для обработки этого файла?		

Л-3	Программное обеспечение	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Программные среды.
- Виртуальные машины Python.
- PVM CPython.
- PVM PyPy.
- Интерактивные оболочки.
- Виртуальные окружения.
- Управление версиями алгоритмов.

Контрольные вопросы:

1. Что относится к известными виртуальными машинами Python?
2. Какие существуют альтернативы CPython? Дайте краткую характеристику наиболее известным.
3. Что понимают под JIT – (Just-in-time compilation)?
4. Что понимают под интерактивной оболочкой языка Python?
5. С помощью каких сервисов производят развертывание интерактивных сред?
6. В чем заключаются назначения виртуального окружения?
7. Охарактеризуйте кроссплатформенный менеджер зависимостей и окружений Conda (Mamba).
8. Что можно отнести к достоинствам и недостаткам Conda (Mamba)?

Л-4	Диагностика алгоритмов	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Интроспекция высокопроизводительных алгоритмов.
- Линтеры и форматеры.
- Тестирование алгоритмов.
- Статистический и сравнительный тайминг.
- Отладка кода.
- Профилирование алгоритмов.

Контрольные вопросы:

1. Назовите основные возможности и назначение интроспекции.
2. В чем заключается назначение линтеров? Приведите примеры наиболее популярных и порядок использования.
3. Что такое автоформатеры и их основное назначение?
4. Приведите примеры и охарактеризуйте известные фреймворки тестирования программного кода языка Python на ошибки.
5. Приведите примеры и охарактеризуйте известные отладчики для Python.
6. Что такое профилирование? Какую информацию предоставляют профайлеры алгоритмов?
7. Какие библиотеки или модули для профилирования алгоритмов вы знаете и в чем заключается их назначение?

Л-5	Параллелизм вычислений	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Понятие «параллелизм вычислений».
- Библиотека Dask.
- Интерфейс распараллеливания.
- Индикатор выполнения задач.
- Профайлер класс.

Контрольные вопросы:

1. Для чего служит библиотека Dask? В чем заключается особенность интеграции Dask и других библиотек?
2. Для чего служит функция **ProgressBar()** библиотеки Dask? Для чего служит класс **Dask.diagnostics** библиотеки Dask?
3. Какую информацию предоставляют классы **Profiler**, **ResourceProfiler** и **CashProfiler** библиотеки Dask?
4. Как визуализировать вычислительный граф? Какие библиотеки и утилиты необходимы для этого?
5. Что такое профилирование? Какую информацию предоставляют профайлеры алгоритмов?
6. Какие библиотеки или модули для профилирования алгоритмов вы знаете и в чем заключается их назначение?

Л-6	Программно-аппаратная акселерация	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Рассматриваются объекты-итераторы и объекты-генераторы.
- Изучается динамическая компиляция и конкурентность вычислений.
- Многопоточные вычисления: глобальная блокировка, вычислительный параллелизм, параллельные циклы, а также внешняя многопоточность.
- Субпроцессорные вычисления и субпроцессорные конвейеры.

Контрольные вопросы:

1. Поясните назначение и основные свойства объектов-итераторов.
2. Что представляют собой объекты-генераторы?
3. Приведите сравнения объектов-итераторов с объектами-генераторами.
4. Какие технологии динамической компиляции вы знаете?
5. Что понимается под конкурентностью вычислений?
6. Поясните принципы работы глобальной блокировки интерпретатора.
7. Раскройте принципы вычислительного параллелизма.
8. Что такое параллельные циклы и в чем заключается их основное назначение?
9. Поясните принципы работы внешней многопоточности и продемонстрируйте на примере совместную работу Dask и Numba.
10. Что понимается под субпроцессными вычислениями в конкурентном программировании? Продемонстрируйте на примере.

Л-7	Мультипроцессные вычисления, контейнерная агрегация и map-reduse	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Пул процессов.
- Идентификаторы процессов.
- Основные методы запуска процессов.
- Интерфейс передачи сообщений.
- MPI коммуникаторы.
- Неблокирующее общение процессов.
- Командный интерфейс MongoDB.
- M-скрипты обработки данных.
- Конвейерная агрегация.
- Map-reduce.

Контрольные вопросы:

1. Какое назначение пула процессов? Приведите пример использования объекта **pool** класса **multiprocessing.**
2. Что такое идентификаторы процессов? Приведите примеры применения.
3. Приведите перечень основных методов запуска процессов. Продемонстрируйте один из них на примере.

4. Поясните понятие интерфейса передачи сообщений и его значения для распараллеливания вычислительных задач для систем с распределенной и общей памятью. Приведите примеры реализации и использования этих интерфейсов.
5. Для чего служат неблокирующие методы общения процессов? Приведите пример распределения подзадач по процессам.
6. В чем заключается отличие операций map и reduce от методов конвейерной агрегации. Приведите пример выполнения операций map и reduce.

Л-8	Нейросетевые акселераторы и распределенные вычисления	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Нейросетевые акселераторы: GPU, TPU, TFL.
- Интерактивные параллельные вычисления.
- Планировщики вычислительных заданий.
- Модели распределенных вычислений.
- Диагностика распределенных вычислений.
- Масштабирование объемных данных.
- Распределенная вычислительная система.

Контрольные вопросы:

1. Использование GPU, TPU и TFL для увеличения производительности вычислительных операций.
2. В чем заключаются основные отличия GPU, TPU и TFL?
3. Каким образом библиотека Ipyparallel расширяет возможности вычислительных систем высокопроизводительно обрабатывать большие объемы данных?
4. В чем заключается основное назначение планировщика вычислительных заданий в распределенных системах?
5. Перечислите и охарактеризуйте модели распределенных вычислений.
6. Назовите наиболее известные интерактивные диагностические панели инструментов планировщиков распределённых вычислений.
7. В чем заключатся смысл масштабирования данных, превышающих размер оперативной памяти?
8. Поясните разницу между распределенной вычислительной системой и суперкомпьютером.
9. Назовите основные виды распределённых вычислительных систем и их классификация по назначению и техническому обеспечению.

Л-9	Вычислительные эксперименты	2 ак. часа

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- Вводится понятие «вычислительного эксперимента»
- Рассматривается обезразмеривание уравнений.
- Изучается программирование модели и ее уровни.
- Вычислительная неустойчивость.
- Версии программных продуктов.

Контрольные вопросы:

1. Раскройте содержимое понятия «вычислительный эксперимент».
2. В чем заключается процесс обезразмеривания уравнений?
3. С помощью чего можно осуществить переход от физической модели к математической?
4. Перечислите уровни моделей вычислительного эксперимента.
5. Какой недостаток алгоритмов вычислительного эксперимента, приводящий к результатам далеким от действительных, наиболее труден для обнаружения и устранения?
6. Как неспециалист в функциональном анализе может обеспечить использование вычислительно устойчивых алгоритмов и программ?

Содержание и темы лабораторных работ представлены в следующей таблице.

ЛР-1	Виртуальные системы	6 ак. часа
Цель выполнения лабораторной работы: *Знакомство с выделенной вычислительной системой выбранной облачной платформы. Знакомство с наиболее популярными виртуальными машинами Python (PVM). Знакомство с интерактивной оболочкой IPython (Interactive Python). Знакомство с виртуальным окружением Python.*		
Результат: *получение практических навыков работы с виртуальными системами.*		
Порядок выполнения лабораторной работы: • Подготовка к выполнению работы, в том числе: • Изучение возможностей по использованию виртуальной выделенной вычислительной системы облачной платформы; • Проведение сравнительного анализа выделенных ресурсов; • Изучение настроек с использование протоколов SSH и SFTP; • Создание пользовательского домена на выделенном ресурсе; • Выбор и установка PVM (виртуальных машин Python) для локальной или виртуальной вычислительной системы; • Использование PVM для проведения простых вычислительных экспериментов; • Изучение проеков Jupyter Lab, Jupyter Notebook и Jupyterhub на базе интерактивной оболочки IPython; • Установка, настройка и использование Jupyter Lab и Jupyter Notebook в локальной или виртуальной вычислительной системе; • Создайте виртуальное окружение для проекта, используя **virtualenv**. • Активация виртуального окружения и установка в проект нескольких библиотек (пакетов) Python. • Подготовка отчета; • Защита лабораторной работы.		
Контрольные вопросы: 1. Что представляет собой виртуальная вычислительная машина? 2. Какие отличительные признаки характеризуют распределенные вычислительные системы? 3. Что такое суперскалярность ядра вычислительной системы? 4. Что представляет из себя параллельная виртуальная машина (PVM)? 5. Перечислите и охарактеризуйте проекты на базе интерактивной оболочки IPython. 6. Что такое виртуальное окружение? 7. Для чего используется модуль **virtualenv**? 8. Перечислите и охарактеризуйте расширения модуля **virtualenv**?		
ЛР-2	Диагностика алгоритмов	6 ак. часа
Цель выполнения лабораторной работы: *Ознакомление с техническими характеристиками операторов и выражений алгоритмов, выполняемых в интерактивной оболочке IPython.*		
Результат: *получение практических навыков измерения времени выполнения, оценки эффективности операторов и выражений алгоритмов, выполняемых в интерактивной оболочке IPython.*		
Порядок выполнения лабораторной работы: • Подготовка к выполнению работы, в том числе: • Выбор литера для анализа программного кода;		

- Проведение проверок используемых алгоритмов;
- Определение времени выполнения оператора или выражения с использованием встроенного в IPython модуля **timeit;**
- Выбор линтера и работа по анализу кода и устранению его ошибок;
- Выбор автоформатера для соответствия стандарту PEP8;
- Работа с интерпретатором для отладки кода;
- Подготовка отчета;
- Защита лабораторной работы.

Контрольные вопросы:

1. Что такое линтеры?
2. Охарактеризуйте линтер Flake8.
3. Какие линтеры Python могут использоваться для редактора Vscode?
4. Что такое автоформатеры?
5. Основное назначение стандарта PEP8.
6. Перечислите и охарактеризуйте известные фреймворки языка программирования Python для тестирования программного кода на ошибки.

ЛР-3	Профилирование алгоритмов	6 ак. часа

Цель выполнения лабораторной работы: *Ознакомление с профилированием с целью оптимизации программ по скорости их выполнения и расходу оперативной памяти*

Результат: *получение навыков профилирования алгоритмов для высокопроизводительных и экспериментальных вычислений*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Создание нового или использование уже существующего виртуального окружения для экспериментов с профилированием алгоритмов;
- Установка пакетов профилирования программного кода **line_profiler**;
- Установка профилировщика памяти **memory_profiler**;
- Исследование всех возможных аргументов команды **mprof run** с различными интервалами измерений;
- Применение команд построения временных диаграмм для визуализации результатов;
- Подготовка отчета;
- Защита лабораторной работы.

Контрольные вопросы:

1. Что такое профилирование?
2. Перечислите и охарактеризуйте типы профилировщиков для языка программирования Python.
3. Характеристика модулей **line_profiler** и **memory_profiler** для профилирования кода.
4. Основное назначение команды **mprof run.**
5. Какие команды применяются для построения временных диаграмм для визуализации результатов?

ЛР-4	Диагностика параллельного кода	6 ак. часа

Цель выполнения лабораторной работы: *Знакомство с диагностическими индикаторами и панелями мониторинга для выполнения сложных и продолжительных по времени вычислений.*

Результат: *получение практических навыков использования диагностических индикаторов и панелей мониторинга*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Изучение популярных методов диагностики выполнения параллельного кода.
- Групповое тестирование нескольких вычислительных функций в одном кластере.
- Анализ работы тестера.
- Установка библиотеки Dask и изучение ее возможностей.
- Изучение процесса визуализации вычислительных графов функцией **visualize().**
- Изучение функции **progressBar()** библиотеки **dask.diagnostics** для отображения индикатора выполнения задачи во время вычислений.
- Изучение инструментов библиотеки Dask для выполнения профилирования.
- Подготовка отчета.
- Защита лабораторной работы.

Контрольные вопросы:

1. Характеристика инструмента VizTracer для параллельных вычислений.
2. Что такое Dask?
3. Какие возможности предоставляет библиотека Dask?
4. Какие возможности предоставляет функция **visualize** библиотеки Dask?
5. Для чего применяется функция **ProgressBar()** библиотеки **dask.diagnostics?**
6. Перечислите и охарактеризуйте инструменты библиотеки Dask для выполнения профилирования.

ЛР-5	Итераторы и генераторы	6 ак. часа

Цель выполнения лабораторной работы: *Ознакомление с объектами-генераторами для выполнения сложных и продолжительных по времени вычислений*

Результат: *получение практических навыков работы с объектами-генераторами*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Использование объектов-генераторов для экспериментальных методов высокопроизводительной обработки данных.
- Использование объектов-итераторов для экспериментальных методов высокопроизводительной обработки данных.
- Решение конкретных задач.
- Подготовка отчета.
- Защита лабораторной работы.

Контрольные вопросы:

1. Какие объекты в Python называют итераторами?
2. В чем заключается основное назначение итераторов?
3. Какие классы в языке программирования Python используются для лучшего понимания принципа работы итераторов?
4. Что такое объекты-генераторы?
5. В чем заключается основное назначение объектов-генераторов и функций-генераторов?
6. Какой класс используется для лучшего понимания структуры объектов-генераторов?

ЛР-6	Динамическая компиляция	6 ак. часа

Цель выполнения лабораторной работы: *Ознакомление с принципами динамической компиляции для выполнения сложных и продолжительных по времени вычислений*

Результат: *получение практических навыков работы применения динамической компиляции для выполнения сложных и продолжительных по времени вычислений*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Изучение пакета Numba с включенным в него JIT-компилятором для языка программирования Python.
- Изучение пакета Numpy для языка программирования Python.
- Изучение возможностей декоратора для трансляции функций в функции самокомпиляции.
- Знакомство с возможностями пакета Numexpr.
- Выполнение на основе сравнительного и статистического тайминга примеров высокопроизводительных вычислений с использованием пакетов Numpy, Numba и Numexpr.
- Подготовка отчета;
- Защита лабораторной работы.

Контрольные вопросы:

1. Что представляет собой **JIT –** (Just-in-time compilation) динамическая компиляция?
2. Охарактеризуйте пакет Numba с включенным в него JIT-компилятором.
3. Для чего используется пакет Numpy?
4. В чем заключаются возможности пакета Numexpr?

ЛР-7	Многопоточность	6 ак. часа

Цель выполнения лабораторной работы: *Ознакомление со способами создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*

Результат: *получение практических навыков создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Изучение модуля **threading** языка программирования Python в качестве примера конкурентного программирования.
- Изучение интерактивной среды Ipython для решения задач многопоточности.
- Реализация собственного алгоритма многопоточности. Выполнение сравнительного тайминга.
- Подготовка отчета.
- Защита лабораторной работы.

Контрольные вопросы:

1. Что понимается под конкурентностью вычислений?
2. Как используется многопоточность для высокопроизводительных вычислений?
3. Поясните понятие и приведите примеры конкурентного программирования.
4. Поясните принципы работы глобальной блокировки интерпретатора.
5. Раскройте принципы вычислительного параллелизма.

ЛР-8	Подпроцессы	6 ак. часа

Цель выполнения лабораторной работы: *Ознакомление со способами создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*

Результат: *получение практических навыков создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Изучение модуля модуля **subprocess** для создания новых процессов и соединения их в конвейеры.
- Изучение интерактивной среды Ipython для выполнения алгоритма запуска конвейера двух подпроцессов.
- Реализация конвейера, передающего список файлов, сформированный одним процессом на вход другого процесса, выполняющего фильтрацию списка по шаблону.
- Реализация собственного алгоритма.
- Подготовка отчета.
- Защита лабораторной работы.

Контрольные вопросы:

1. Что понимается под субпроцессными вычислениями в конкурентном программировании? Продемонстрируйте на примере.
2. Объясните работу субпроцессных конвейеров. Продемонстрируйте пример использования.
3. Поясните основные методы запуска процессов.
4. В чем заключается использование неблокирующих методов **isend()** и **irecv()**?

ЛР-9	Мультипроцессинг	6 ак. часа

Цель выполнения лабораторной работы: *Ознакомление со способами создания программных продуктов на основе мультипроцессинга*

Результат: *получение практических навыков создания программных продуктов на основе мультипроцессинга*

Порядок выполнения лабораторной работы:

- Подготовка к выполнению работы, в том числе:
- Изучение пакета **multiprocessing** для выполнения многопроцессорных вычислений с использованием API.
- Изучение объекта **Pool** для параллелизма данных.
- Изучение метода **Process** для вывода отдельных идентификаторов процессов
- Подготовка отчета.
- Защита лабораторной работы.

Контрольные вопросы:

1. В чем заключается основное назначение модуля **multiprocessing?**
2. Для чего используется объект **Pool**?
3. Какой объект отвечает за создание процессов в многопроцессорной обработке с последующим вызовом метода **start()**?
4. Сколько способов запуска процесса поддерживает модуль **multiprocessing**?
5. На что ориентирован интерфейс передачи сообщений (MPI)?
6. Какой стандарт используется для распараллеливания программ**?**

Календарный график дисциплины

№	Раздел	Недели	Виды учебной работы, ак. часы					Форма промежуточной аттестации
			Лекции	Семинары	Лабораторные работы	Консультации	Самостоятельная работа	
Первый семестр изучения дисциплины								
1	**Лекция Л-1.** *Основные понятия и законы высокопроизводительных вычислений*	1	2				6	
2	**Лекция Л-2.** *Стек вычислительных технологий*	2	2				6	
3	**Лекция Л-3.** *Программное обеспечение*	3	2				6	
4	**Лекция Л-4.** *Диагностика алгоритмов*	4	2				6	
5	**Лекция Л-5.** *Параллелизм вычислений*	5	2				6	
6	**Лекция Л-6.** *Программно-аппаратная акселерация*	6	2				6	
7	**Лекция Л-7.** *Мультипроцессные вычисления, контейнерная агрегация и map-reduse*	7	2				6	
8	**Лекция Л-8.** *Нейросетевые акселераторы и распределенные вычисления*	8	2				6	
9	**Лекция Л-9.** *Вычислительные эксперименты*	9	2				6	
10	**Лабораторная работа ЛР-1** *Виртуальные системы*				6		10	
11	**Лабораторная работа ЛР-2** *Диагностика алгоритмов*				6		10	
12	**Лабораторная работа ЛР-3** *Профилирование алгоритмов*				6		10	
13	**Лабораторная работа ЛР-4** *Диагностика параллельного кода*				6		10	
14	**Лабораторная работа ЛР-5** *Итераторы и генераторы*				6		10	
15	**Лабораторная работа ЛР-6** *Динамическая компиляция*				6		10	
16	**Лабораторная работа ЛР-7** *Многопоточность*				6		10	
17	**Лабораторная работа ЛР-8**				6		10	

	Подпроцессы							
18	**Лабораторная работа ЛР-9** *Мультипроцессинг*				6		10	
	Промежуточная аттестация							З
	Итого в семестре:		**18**		**54**		**144**	
	ИТОГО по дисциплине:		**18**		**54**		**144**	

4. Образовательные технологии

Методика преподавания дисциплины «Высокопроизводительные вычисления» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

- подготовка к лекциям и к выполнению практических работ;
- выполнение лабораторных работ;
- использование интерактивных форм проведения занятий.

Удельный вес занятий, проводимых в интерактивных формах, определен образовательной программой, особенностью контингента обучающихся и содержанием дисциплины «Высокопроизводительные вычисления» и в целом по дисциплине составляет 37% аудиторных занятий. Занятия лекционного типа составляют примерно 13% от объема аудиторных занятий.

Самостоятельная внеаудиторная работа студентов составляет 63% от общего объема дисциплины и состоит из:

- повторения и систематизации лекционного материала;
- чтения литературы и освоения дополнительного материала в рамках тематики дисциплины;
- подготовки к текущей аттестации;
- подготовки к промежуточной аттестации.

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

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

В седьмом семестре

Проверка домашних заданий
Проверка готовности студентов
Проверка выполнения лабораторных работ.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.
В результате освоения дисциплины формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-1	способен разрабатывать и адаптировать прикладное программное обеспечение

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин, практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины, описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ПК-1 способен разрабатывать и адаптировать прикладное программное обеспечение				
Показатель:	**Критерии оценивания**			
	Допороговое значение	**Пороговое значение**		
	2	**3**	**4**	**5**

Знать: требования к программному коду; языки програм-мирования, определения и манипулирования данными.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.
Уметь: составлять программный код с использованием языков программирования, определения и манипулирования данными; производить проверку и отладку программного кода; работать с системой контроля версий; разрабатывать и документировать программный интерфейс; разрабатывать технические спецификации на программные компоненты и их взаимодействие; разрабатывать тестовые наборы данных и процедур проверки работоспособности программного обеспечения; осуществлять интеграцию программных модулей и	Обучающийся не умеет или в недостаточной степени умеет выполнять действия, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3).	Обучающийся демонстрирует неполное соответствие умений, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность умений, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие умений, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3). Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся демонстрирует полное соответствие умений, указанных в индикаторах компетенций дисциплины «Уметь» (см. п. 3). Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.

компонент и верификации выпусков программного продукта: составлять формализованные описания решений, поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных.				
Владеть: навыками оптимизации программного кода; навыками проверки работоспособности программного обеспечения; навыками анализа возможностей реализации требований к программному обеспечению; навыками проектирования программного обеспечения; навыками исправления дефектов, зафиксированных в базе данных.	Обучающийся не владеет или в недостаточной степени владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3).	Обучающийся в неполном объеме владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность владения навыками по ряду показателей. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3). Навыки освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме владеет приемами, методами и иными умениями, указанными в индикаторах компетенций дисциплины «Владеть» (см. п. 3). Свободно применяет полученные навыки в ситуациях повышенной сложности.

Образцы контрольных вопросов и заданий для проведения текущего контроля, приведены в приложении 2.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Промежуточная аттестация обучающихся в форме зачета проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «зачтено», «не зачтено».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Высокопроизводительные вычисления» (указывается что именно – прошли промежуточный контроль, выполнили лабораторные работы, выступили с докладом и т.д.)

Шкала оценивания	Описание
Зачтено	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Не зачтено	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Фонды оценочных средств представлены в приложении 2 к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная литература

1. Евтихов В.Г. и др., Высокопроизводительные вычисления. - Учебник для вузов. – М.: Изд-во Московский политехнический университет, 2022. - 323 с.

7.2. Дополнительная литература

1. Евтихов В.Г. Акселерация обработки данных: учебно-методическое пособие / В.Г. Евтихов, Н.В. Евтихова, С.В. Суворов. – Москва: Московский Политех, 2021. —137 с.

8. Материально-техническое обеспечение дисциплины.

8.1 Требования к оборудованию и помещению для занятий

Лекционные занятия могут проводиться как в специализированных аудиториях с комплектом мультимедийного оборудования, так и с использованием средств дистанционного обучения. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

8.2 Требования к программному обеспечению

Для проведения практических занятий необходимо специальное программное обеспечение:

- многоядерные вычислительные системы на базе операционной системы Linux с возможностью удаленного доступа по протоколам SSH/SFTP;
- предустановленная PVM (CPython, PyPy);
- доступ в интернет для установки дополнительного программного обеспечения из общедоступных репозиториев.

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются *аудиторные занятия, лекции.*

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на практических занятиях, промежуточный контроль осуществляется на зачете.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

10. Методические рекомендации для преподавателя

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

3. При организации и проведения экзаменов в практико-ориентированной форме следует использовать утвержденные кафедрой Методические рекомендации.

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Направление подготовки:

09.03.03 «Прикладная информатика»

Форма обучения: очная

Вид профессиональной деятельности: (В соответствии с ФГОС ВО)

Кафедра: Прикладная информатика

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ «Высокопроизводительные вычисления»

Состав:

1. Паспорт фонда оценочных средств
2. Перечень оценочных средств
3. Оценочные средства

Москва 2022 г.

Паспорт фонда оценочных средств по дисциплине «Высокопроизводительные вычисления» по направлению подготовки 09.03.03 «Прикладная информатика» (бакалавр).

«Высокопроизводительные вычисления»					
ФГОС ВО 09.03.03 «Прикладная информатика» (уровень бакалавриата)					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **профессиональные компетенции**:					
Компетенции		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства**	**Степени уровней освоения компетенций**
Индекс	**Формулировка**				
ПК-1	способен разрабатывать и адаптировать прикладное программное обеспечение	**Знать:** требования к программному коду; языки программирования, определения и манипулирования данными. **Уметь:** составлять программный код с использованием языков программирования, определения и манипулирования данными; производить проверку и отладку программного кода; работать с системой контроля версий; разрабатывать и документировать программный интерфейс; разрабатывать технические спецификации на программные компоненты и их взаимодействие;	лекции, лабораторные работы.	Зачет	**пороговый уровень:** Обучающийся способен сформулировать 1 задачу, в которой применяются высокопроизводительные вычислительные системы и способен сформулировать основные принципы и основы проведения многопроцессных и распределенных вычислений. **базовый уровень:** Обучающийся способен сформулировать несколько задач, в которых применяются высокопроизводительные вычислительные системы. **повышенный уровень:** Обучающийся способен

		разрабатывать тестовые наборы данных и процедур проверки работоспособности программного обеспечения; осуществлять интеграцию программных модулей и компонент и верификации выпусков программного продукта: составлять формализованные описания решений, поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных. **Владеть:** навыками оптимизации программного кода; навыками проверки работоспособности программного обеспечения; навыками анализа возможностей реализации требований к программному обеспечению; навыками проектирования программного обеспечения; навыками исправления дефектов, зафиксированных в базе данных.			сформулировать несколько задач, в которых применяются высоко-производительные вычислительные систе-мы, а также основы программирования на графических процес-сорах. Разработать и отладить программу с использованием программирования на графических процессорах

Примерный перечень вопросов к зачету по курсу «Высокопроизводительные вычисления»:

1. Где используются и применяются методы высокопроизводительной обработки данных?
2. Какой показатель является существенным для вычислительных систем?
3. Какую систему принято называть вычислительным кластером Beowulf?
4. Поясните назначение понятия «флопс».
5. Что используют для определения производительности компьютеров?
6. В чем заключается закон Амдала?
7. В чем заключается закон Густавсона-Барсиса?
8. Раскройте содержание закона Мура и проанализируйте его.
9. Что такое суперкопьютеры?
10. В чем состоит назначение вычислительного кластера?
11. Назовите основные отличия суперЭВМ от майнфрема.
12. Какой проект публикует актуальный перечень суперкомпьютеров?
13. Какой компьютер является лидером среди суперкомпьютеров?
14. Назовите основные характеристики суперкомпьютера Фугаку.
15. Перечислите области применения суперкомпьютеров?
16. На чем основывается стек вычислительных технологий?
17. Что понимается под распределенными вычислениями?
18. Дайте общее определение масштабируемости вычислительной системы?
19. Что вы понимаете под горизонтальным масштабированием вычислительной системы?
20. Что вы понимаете под вертикальным масштабированием вычислительной системы?
21. Поясните способ организации параллельных вычислений.
22. Что относится к источникам больших наборов данных?
23. Дайте определение понятию «большие данные».
24. Дайте характеристику облачным вычислениям.
25. Дайте характеристику основным моделям обслуживания.
26. Что представляет собой виртуальная вычислительная машина?
27. Дайте определение понятию «туманные вычисления».
28. В чем заключаются отличии туманных вычислений от облачных?
29. Дайте определение понятию «граничные вычисления».
30. Какие требования предъявляют к граничным вычислениям?
31. Охарактеризуйте распределенные вычисления.
32. Дайте несколько определений понятию «распределенная вычислительная система».
33. Какие отличительные признаки характеризуют распределенные вычислительные системы.
34. Что такое суперскалярность ядра вычислительной системы?
35. В каком виде может быть представлена визуализация параллелизма?
36. Приведите краткую характеристику платформы Apache Hadoop.
37. Является ли платформа Hadoop монолитной (неделимой), обоснуйте свой ответ.

38. В чем заключаются особенности файловой системы HDFS?

39. Опишите алгоритмы технологий обработки данных MapReduce.

40. Известно, что в файле конфигурации Hadoop задан размер блока 64 МБ. Файл bigdata.gz занимает 1 ГБ места в HDFS. Данный файл был выбран для некой MapReduce операции. Сколько мапперов будет создано для обработки этого файла?

41. Для чего служит интерфейс передачи сообщений (Message Passing Interface, MPI)? Проанализируйте схему работы.

42. В чем состоит назначение параллельной виртуальной машины (Parallel Virtual Machine, PVM)? Проанализируйте схему работы.

43. Что понимают под единой операционной системой (Single system image, SSI)?

44. Поясните основные отличия и достоинства компиляторов и интерпретаторов.

45. Назовите основные типы интерпретаторов.

46. Назовите основное назначение трансляции программы в байт-код.

47. Что такое PVM и как не путать параллелизм с интерпретацией?

48. Что относится к известными виртуальными машинами Python?

49. Назовите и охарактеризуйте наиболее распространённую, эталонную реализацию языка программирования Python.

50. Что относится к актуальной задаче для высокопроизводительных вычислений?

51. Что является альтернативой CPython?

52. Что понимают под JIT – (Just-in-time compilation)?

53. Что понимают под интерактивной оболочкой языка Python?

54. В чем заключается параллелизм IPython?

55. Раскройте уникальные характеристики IPython?

56. Охарактеризуйте веб-приложение с открытым исходным кодом Jupyter Notebook.

57. Поясните достоинства и недостатки Jupyter Lab в сравнении с Jupyter Notebook.

58. Назовите и охарактеризуйте многопользовательскую версию Jupyter Hub.

59. С помощью каких сервисов производят развертывание интерактивных сред?

60. Для чего используются виртуальные окружения?

61. Виртуальное окружение – это?

62. В чем заключаются назначения виртуального окружения?

63. Что такое Virtualenv и для чего используется?

64. Какие команды нужно выполнить для создания виртуального окружения?

65. Что можно отнести к недостаткам virtualenv?

66. Сколько и какие способы зафиксировать версии пакетов в файл существуют?

67. Что такое VirtualEnvWrapper?

68. Охарактеризуйте кроссплатформенный менеджер зависимостей и окружений Conda.

69. Что можно отнести к достоинствам и недостаткам Conda?

70. Что такое Pipenv?

71. Какие проблемы virtualenv и conda позволяет решить Pipenv?

72. Что представляет из себя синтаксис Pipfile?

73. Приведите пример и дайте характеристику системы отслеживания дефектов и изменений?

74. Что является текстовым форматом, предназначенным для обмена данными в среде Jupyter? В чем заключаются его возможности?

75. Назовите основные возможности и назначение интроспекции.

76. Каково назначение модуля sys?

77. В чем заключается назначение линтеров? Приведите пример программы.

78. Какой линтер Python можно использовать для редактора Vscode?

79. Что такое автоформатеры?

80. В чем заключается основная задача стандарта PEP8?

81. Приведите примеры и охарактеризуйте известные фреймворки тестирования программного кода языка Python на ошибки.

82. Какие команды интерактивной оболочки IPython можно использовать для измерения производительности программного кода?

83. Что такое тайминг? Какой оператор отвечает за тайминг в IPython?

84. Какие из опций timeit являются наиболее используемыми?

85. Какие опции оператора %xmode позволяют контролировать объем выводимой информации при возникновении исключения?

86. Перечислите команды, использующиеся для интерактивной отладки.

87. Приведите примеры и охарактеризуйте известные отладчики для Python.

88. Что такое профилирование? Какую информацию предоставляют профайлеры алгоритмов?

89. Какие библиотеки или модули для профилирования алгоритмов вы знаете и в чем заключается их назначение?

90. Для чего служит библиотека Dask в параллельных вычислениях?

91. В чем заключается особенность интеграции Dask и других библиотек?

92. Как визуализировать вычислительный граф? Какие библиотеки и утилиты необходимы для этого?

93. Для чего служит функция ProgressBar() библиотеки Dask?

94. Для чего служит класс Dask.diagnostics библиотеки Dask?

95. Какую информацию предоставляют классы Profiler, ResourceProfiler и CashProfiler библиотеки Dask?

96. Какую информацию могут предоставить интерактивные диаграммы Bokeh при использовании их с Dask?