Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 13.08.2024 17:51:24
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____/ Д.Г. Демидов /
«_28_» _____марта_____ 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Высокопроизводительные вычисления»

Направление подготовки/специальность
09.03.03 Прикладная информатика

Профиль/специализация
Большие и открытые данные

Квалификация
бакалавр

Формы обучения
заочная

Москва, 2024 г.

Разработчик(и):

К. т. н., доцент / В.Г. Евтихов /

Согласовано:
Заведующий кафедрой «Прикладная информатика»,
К. э. н., доцент /С.В.Суворов /

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине....4
2 Место дисциплины в структуре образовательной программы....5
3 Структура и содержание дисциплины....5
3.1 Виды учебной работы и трудоемкость....5
3.2 Тематический план изучения дисциплины....6
3.3 Содержание дисциплины....8
3.4 Тематика семинарских/практических и лабораторных занятий....9
3.5 Тематика курсовых проектов (курсовых работ)....10
4 Учебно-методическое и информационное обеспечение....10
4.1 Нормативные документы и ГОСТы....10
4.2 Основная литература....10
4.3 Дополнительная литература....11
4.4 Электронные образовательные ресурсы....11
4.5 Лицензионное и свободно распространяемое программное обеспечение....11
4.6 Современные профессиональные базы данных и информационные справочные системы....11
5 Материально-техническое обеспечение....11
5.1 Требования к оборудованию и помещению для занятий....11
5.2 Требования к программному обеспечению....11
6 Методические рекомендации....11
6.1 Методические рекомендации для преподавателя по организации обучения....11
6.2 Методические указания для обучающихся по освоению дисциплины....12
7 Фонд оценочных средств....12
7.1 Методы контроля и оценивания результатов обучения....12
7.2 Шкала и критерии оценивания результатов обучения....12
7.3 Оценочные средства....16

1 Цели, задачи и планируемые результаты обучения по дисциплине

Целями дисциплины являются получение представления о современных подходах, используемых при создании высокопроизводительных 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.

Обучение по дисциплине «Высокопроизводительные вычисления» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
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. 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 документов. **Владеть**: Навыками оптимизации программного кода. Навыками проверки работоспособности программного обеспечения. Навыками анализа

	возможностей реализации требований к программному обеспечению. Навыками проектиорвания программного обеспечения. Навыками исправления дефектов, зафиксированных в базе данных.

2 Место дисциплины в структуре образовательной программы

Дисциплина «Высокопроизводительные вычисления» относится к элективным дисциплинам (элективные дисциплины №6) части (части, формируемой участниками образовательных отношений) блока Б1 «Дисциплины (модули)».

Входными требованиями, необходимыми для освоения дисциплины, является наличие у обучающихся компетенций, сформированных при изучении дисциплин:

- «Программирование»;
- «Теоретические основы информатики»;
- «Основы баз данных».

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единицы, т.е. 216 академических часа (из них 192 часов – самостоятельная работа студентов).

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

Лекции – 8 часов, лабораторные работы – 16, форма контроля – экзамен.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Заочная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			8	
1	**Аудиторные занятия**	**54**	54	
	В том числе:			
1.1	Лекции	18	8	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	36	16	
2	**Самостоятельная работа**	**192**	192	
	В том числе:			
2.1	Лабораторные работы	192	192	
2.2	…			
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен		Экзамен	
	Итого:	**216**	216	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Заочная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Раздел 1.						
1.1	Тема 1. ОСНОВНЫЕ ПОНЯТИЯ	7	1				6
1.2	Тема 2. СТЕК ВЫЧИСЛИТЕЛЬНЫХ ТЕХНОЛОГИЙ	7	1				6
1.3	Тема 3. РАСПРЕДЕЛЁННОСТЬ И ПАРАЛЛЕЛЬНОСТЬ	7	1				6
1.4	Тема 4. СУПЕРКОМПЬЮТЕРЫ	7	1				6
1.5	Тема 5. ВИЗУАЛИЗАЦИЯ ПАРАЛЛЕЛИЗМА	7	1				6
1.6	Тема 6. ПРОГРАММНЫЕ СРЕДЫ	7	1				6
1.7	Тема 7. ДИАГНОСТИКА АЛГОРИТМОВ	7	1				6
1.8	Тема 8. ПАРАЛЛЕЛИЗМ ВЫЧИСЛЕНИЙ	7	1				6
1.9	Тема 9. ПРОГРАММНО-АППАРАТНАЯ АКСЕЛЕРАЦИЯ	7					7
2	Раздел 2						
2.1	Лабораторная работа 1 Виртуальные системы *Знакомство с: выделенной вычислительной системой выбранной облачной платформы; наиболее популярными виртуальными машинами Python (PVM); интерактивной оболочкой IPython (Interactive Python); виртуальным окружением Python.*	17			2		15
2.2	Лабораторная работа 2 Диагностика алгоритмов *Знакомство с техническими характеристиками операторов и выражений алгоритмов, выполняемых в интерактивной оболочке IPython.*	17			2		15
2.3	Лабораторная работа 3 Профилирование алгоритмов	17			2		15

	Ознакомление с профилированием с целью оптимизации программ по скорости их выполнения и расходу оперативной памяти						
2.4	Лабораторная работа 4 Диагностика параллельного кода *Знакомство с диагностическими индикаторами и панелями мониторинга для выполнения сложных и продолжительных по времени вычислений.*	17			2		15
2.5	Лабораторная работа 5 Итераторы и генераторы *Ознакомление с объектами-генераторами для выполнения сложных и продолжительных по времени вычислений*	17			2		15
2.6	Лабораторная работа 6 Динамическая компиляция *Ознакомление с принципами динамической компиляции для выполнения сложных и продолжительных по времени вычислений*	17			2		15
2.7	Лабораторная работа 7 Многопоточность *Ознакомление со способами создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*	17			2		15
2.8	Лабораторная работа 8 Подпроцессы *Ознакомление со способами создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*	17			1		16
2.9	Лабораторная работа 9 Мультипроцессинг *Ознакомление со способами создания программных продуктов на основе мультипроцессинга*	17			1		16
Итого		**216**	**8**		**16**		**192**

3.3 Содержание дисциплины

Раздел 1.

Тема 1. ОСНОВНЫЕ ПОНЯТИЯ

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- *Параллелизм и производительность*
- *Закон Амдала*
- *Закон Густавсона – Барсиса*
- *Закон Мура*

Тема 2. СТЕК ВЫЧИСЛИТЕЛЬНЫХ ТЕХНОЛОГИЙ

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.*

Тема 3. РАСПРЕДЕЛЁННОСТЬ И ПАРАЛЛЕЛЬНОСТЬ

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.*

Тема 4. СУПЕРКОМПЬЮТЕРЫ

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- *Рассматривается прогресс суперкомпьютеров*
- *Изучается динамическая компиляция и конкурентность вычислений.*
- *Многопоточные вычисления: глобальная блокировка, вычислительный параллелизм, параллельные циклы, а также внешняя многопоточность.*

Тема 5. ВИЗУАЛИЗАЦИЯ ПАРАЛЛЕЛИЗМА

Краткое содержание (перечень рассматриваемых вопросов) лекции:

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

Тема 6. ПРОГРАММНЫЕ СРЕДЫ

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- *Виды программных сред.*
- *Виртуальные машины Python.*
- *PVM CPython.*
- *PVM PyPy.*
- *Интерактивные оболочки.*
- *Виртуальные окружения.*
- *Управление версиями алгоритмов.*

Тема 7. ДИАГНОСТИКА АЛГОРИТМОВ

Краткое содержание (перечень рассматриваемых вопросов) лекции:

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

- *Профилирование алгоритмов.*

Тема 8. ПАРАЛЛЕЛИЗМ ВЫЧИСЛЕНИЙ

Краткое содержание (перечень рассматриваемых вопросов) лекции:

- *Субпроцессорные вычисления и субпроцессорные конвейеры.*
- *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.*

Тема 9. ПРОГРАММНО-АППАРАТНАЯ АКСЕЛЕРАЦИЯ

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.*

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

Семинарские/практические занятия не предусмотрены учебным планом.

3.4.2 Лабораторные занятия

№	Темы лабораторных работ	Часы
1	Виртуальные системы *Знакомство с:* *выделенной вычислительной системой выбранной облачной платформы; наиболее популярными виртуальными машинами Python (PVM); интерактивной оболочкой IPython (Interactive Python); виртуальным окружением Python.*	2
2	Диагностика алгоритмов *Знакомство с техническими характеристиками операторов и выражений алгоритмов, выполняемых в интерактивной оболочке IPython.*	2
3	Профилирование алгоритмов *Ознакомление с профилированием с целью оптимизации программ по скорости их выполнения и расходу оперативной памяти*	2
4	Диагностика параллельного кода *Знакомство с диагностическими индикаторами и панелями мониторинга для выполнения сложных и*	2

	продолжительных по времени вычислений.	
5	Итераторы и генераторы *Ознакомление с объектами-генераторами для выполнения сложных и продолжительных по времени вычислений*	2
6	Динамическая компиляция *Ознакомление с принципами динамической компиляции для выполнения сложных и продолжительных по времени вычислений*	2
7	Многопоточность *Ознакомление со способами создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*	2
8	Подпроцессы *Ознакомление со способами создания программных продуктов, допускающих одновременное выполнение нескольких вычислительных потоков, которые могут взаимодействовать друг с другом.*	1
9	Мультипроцессинг *Ознакомление со способами создания программных продуктов на основе мультипроцессинга*	1

3.5 Тематика курсовых проектов (курсовых работ)

Не предусмотрено

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный государственный образовательный стандарт высшего образования (уровень бакалавриата) по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 №922 «Об утверждении федерального государственного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.03 Прикладная информатика. https://fgos.ru/fgos/fgos-09-03-03-prikladnaya-informatika-922/

2. ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления. Применяется с 01.07.2018. Заменяет ГОСТ 7.32-2001

4.2 Основная литература

1. Евтихов В.Г. и др., Высокопроизводительные вычисления. - Учебник для вузов. – М.: Изд-во Московский политехнический университет, 2024. - 323 с.

4.3 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.

4.4 Электронные образовательные ресурсы

1. Курс: Высокопроизводительные вычисления
https://online.mospolytech.ru/course/view.php?id=13203

2. Учебные материалы в онлайн формате на GitHub
https://vlhub.github.io/HpcUp/html/index.html

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Операционные системы Linux
2. Свободно распространяемые версии Python

4.6 Современные профессиональные базы данных и информационные справочные системы

1. *Нереляционная, NoSQL, документо-ориентированная система управления базами данных MongoDB.*

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

5.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

1. Операционные системы Linux
2. Свободно распространяемые версии Python
3. ПО, предоставленное преподавателем.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

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.

6.2 Методические указания для обучающихся по освоению дисциплины

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

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, 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 аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

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.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

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

Лабораторные работы, зачет.

7.2 Шкала и критерии оценивания результатов обучения

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

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ПК-1. Способен разрабатывать и адаптировать прикладное программное обеспечение.				
ПК-1.1. Знать:	Обучающийся	Обучающийся	Обучающийся	Обучающийся

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	демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	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.	демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов. 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.				

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

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

Шкала оценивания	**Описание**
Отлично	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при переносе знаний и умений на новые, нестандартные задачи.
Хорошо	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. В то же время при ответе допускает несущественные погрешности, задачи решает с недочетами, не влияющими на общий ход решения.
Удовлетворительно	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. Но показывает неглубокие знания, при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами, в решении задач могут содержаться грубые ошибки. Для получения правильного ответа требуются уточняющие вопросы.
Неудовлетворительно	Не выполнены обязательные **условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины, ИЛИ студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями.

7.3 Оценочные средства

7.3.1 Текущий контроль

Проверка и защита лабораторных работ

7.3.2 Промежуточная аттестация

Примерный перечень вопросов к экзамену 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 вычисли-тельные 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?