

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики

**УТВЕРЖДАЮ**  
декан факультета вычислительной  
математики и кибернетики

**И.А. Соколов /**  
**«27» сентября 2023г.**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине

**Облачные и туманные вычисления**

**Уровень высшего образования:**

**бакалавриат**

**Направление подготовки / специальность:**

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

**Направленность (профиль) ОПОП:**

**Искусственный интеллект и анализ данных**

**Форма обучения:**

**очная**

Рассмотрен и утвержден

*на заседании Ученого совета факультета ВМК*

*(протокол №7, от 27 сентября 2023 года)*

Москва 2023

## 1. ФОРМЫ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

В процессе и по завершении изучения дисциплины оценивается формирование у студентов следующих компетенций:

Планируемые результаты обучения по дисциплине (модулю)		
Содержание и код компетенции.	Индикатор (показатель) достижения компетенции	Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций
ПК-8. Способен разрабатывать системы анализа больших данных	<p>ПК-8.1. Разрабатывает программные компоненты извлечения, хранения, подготовки больших данных с учетом вариантов использования больших данных, определений, словарей и эталонной архитектуры больших данных</p> <p>ПК-8.2. Разрабатывает программные компоненты обработки, удаленной, распределенной и объединенной аналитики, использования результатов анализа, описания и управления качеством и достоверностью больших данных</p>	<p>ПК-8.1. 3-1. Знает общедоступные репозитории и специализированные библиотеки, содержащие наборы больших данных</p> <p>ПК-8.1. 3-2. Знает принципы работы экосистемы Hadoop, фреймворка SPARK</p> <p>ПК-8.1. 3-3. Знает устройство интерфейсов между реляционными SQL-хранилищами данных и нереляционными NoSQL-хранилищами данных</p> <p>ПК-8.1. 3-4. Знает предметно-ориентированные языки</p> <p>ПК-8.1. У-1. Умеет настраивать и оптимизировать конфигурацию программного и аппаратного обеспечения с целью интеграции больших данных</p> <p>ПК-8.1. У-2. Умеет разрабатывать программное обеспечение для очистки и валидации наборов больших данных</p> <p>ПК-8.1. У-3. Умеет выполнять потоковую обработку данных (data streaming, event processing)</p> <p>ПК-8.1. У-4. Умеет использовать шины данных (Apache Kafka)</p> <p>ПК-8.1. У-5. Умеет использовать языки запросов, в том числе нереляционных, для поддержки различных типов данных (например, XML, RDF, JSON, мультимедиа) и операций с большими</p>

		<p>данными (например, матричные операции)</p> <p>ПК-8.2. 3-1. Знает принципы и методы анализа больших данных, включая спецификации и стандартизацию метаданных</p> <p>ПК-8.2. 3-2. Знает устройство и принципы работы систем обработки и анализа больших массивов данных (SQL, NoSQL, Hadoop, ETL)</p> <p>ПК-8.2. 3-3. Знает архитектуру и принципы работы промышленных решений, созданных на основе искусственного интеллекта</p> <p>ПК-8.2. 3-4. Знает методы и технологии машинного обучения на больших данных</p> <p>ПК-8.2. У-1. Умеет разрабатывать программное обеспечение для анализа больших данных.</p> <p>ПК-8.2. У-2. Умеет разрабатывать программные и технические средства визуализации больших данных и результатов их анализа.</p> <p>ПК-8.2. У-3. Умеет использовать системы обработки и анализа больших массивов данных (SQL, NoSQL, Hadoop, ETL процессы и инструменты)</p> <p>ПК-8.2. У-4. Умеет использовать технологии науки о данных и больших данных в разработке для решения практических задач промышленности</p> <p>ПК-8.2. У-5. Умеет описывать и управлять качеством и достоверностью больших данных</p>
--	--	---

### 1.1. Текущий контроль успеваемости

Текущий контроль успеваемости осуществляется путем оценки результатов выполнения заданий практических (семинарских) занятий, самостоятельной работы, предусмотренных учебным планом и посещения занятий/активность на занятиях.

В качестве оценочных средств текущего контроля успеваемости предусмотрены:

### **Примерные практические контрольные задания для текущего контроля успеваемости.**

1. Установить гипервизор kvm (инструкция по установке <http://help.ubuntu.ru/wiki/kvm>)
  2. Скачать образ с ОС Linux (<http://mirror.yandex.ru/>) и средствами CLI kvm запустить установку ОС. Полная документация по всем аргументам находится по адресу (<https://qemu.weilnetz.de/doc/qemu-doc.html>)
    - a. Перед созданием виртуальной машины жесткий диск должен быть создан, а ISO образ установочного диска должен находиться по нужному адресу.  
Пример создания жесткого диска  
`qemu-img create -f qcow2 ${HDD_PATH} 30G`  
Подключение к консоли виртуальной машины:  
`spicy —uri=spice://localhost:{YOURPORT}`
  3. После установки операционной системы на основе CLI и аргументов с которыми вы запустили ОС, создать файл xml для virsh (Примеры [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Virtualization\\_Administration\\_Guide/sub-section-Domain\\_Commands-Converting\\_QEMU\\_arguments\\_to\\_domain\\_XML.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/sub-section-Domain_Commands-Converting_QEMU_arguments_to_domain_XML.html)) и из созданной xml создать виртуальную машину средствами virsh и запустить. Запущенную через kvm машину можно погасить
  4. Совместно с другим вариантом, объедините виртуальные машины в один L2 сегмент с помощью VXLAN. (Топология соединения представлена на рисунке рис.1)
- 
1. Создать виртуальную машину удобным для Вас способом с ОС Linux.
  2. Установить средство контейнерной виртуализации Docker ([https://www.docker.com/products/overview#/install\\_the\\_platform](https://www.docker.com/products/overview#/install_the_platform)); установить OpenFlow контроллер RunOS (<https://github.com/ARCCN/runos>); установить средство эмуляции сетей Mininet (<http://mininet.org/>).
  3. Должно получиться 4 docker контейнера:
    - a. В одном докер-контейнере запущен мининет с сетью, имеющей топологию full-mesh, состоящей из 5 свитчей. Эти свитчи подключены к OpenFlow-контроллеру
    - b. В другом докер-контейнере запущен OpenFlow-контроллер. Веб-интерфейс контроллера должен быть доступен на 8080 порту host машины.
    - c. В 3 контейнере должен быть apache и phpmyadmin доступный по 80 порту, файлы тестового сайта расположены на host машине.
    - d. В 4 контейнере должен быть установлен mysql с любыми данными.
  4. На рисунке 1 представлена топология. База данных mysql должна быть доступна через свитчи созданные в mininet.

## 1.2. Промежуточная аттестация

Промежуточная аттестация осуществляется в форме экзамена

В качестве средств, используемых на промежуточной аттестации предусматривается:

Билеты

## 1.3. Типовые задания для проведения промежуточной аттестации

### Список вопросов

1. Перечислите подходы виртуализации. Опишите их особенности, отличия.
2. Перечислите виды услуг, предоставляемые облаком. Опишите их особенности, отличия
3. Перечислите модели размещения облаков. Приведите краткое описание каждой модели.
4. Дайте определение контейнерной виртуализации. Опишите систему виртуализации Docker, основные модули.
5. Дайте определение контейнерной виртуализации. Опишите процесс создания контейнера в системе Docker, основные модули.
6. Дайте определение гипервизора, перечислите типы гипервизоров, приведите примеры гипервизоров
7. Дайте определение SaaS. Опишите преимущества, приведите примеры.
8. Дайте определение PaaS. Опишите преимущества, приведите примеры.
9. Дайте определение IaaS. Опишите преимущества, приведите примеры.
10. Дайте определение динамической трансляции. Опишите основные проблемы при использовании динамической трансляции
11. Опишите основные виды масштабирования облачного сервиса. Перечислите проблемы каждого вида масштабируемости.
12. Перечислите и дайте краткое описание основных компонентов OpenStack
13. Дайте определение контейнерной виртуализации. Опишите роль Dockerfile в процессе определения контейнера.
14. Дайте определение балансировки нагрузки в облачной инфраструктуре. Приведите различные типы балансировщиков нагрузки, опишите их достоинства и недостатки.
15. Архитектура облачного приложения. Свойства. Приложения для аналитики данных.
16. Архитектура облачного приложения. Свойства. Приложения для доставки контента.
17. Сервис-ориентированная архитектура (SOA). Определение. Верхнеуровневая архитектура.
18. Методология проектирования приложений Cloud Component Model. Основные особенности.
19. Язык спецификации облачных приложений Topology and Orchestration Specification for Cloud Applications (TOSCA). Основной синтаксис. Состав TOSCA-шаблона.
20. Виртуализация сетевых функций. Определение. Основные компоненты.
21. Сетевая функция и сетевой сервис. Определение. Примеры.
22. Жизненный цикл виртуального сетевого сервиса. Основные стадии.
23. Вариант использования виртуализации сетевых функций vCPE.
24. Бенчмаркинг облачного приложения. Жизненный цикл.
25. Подходы для генерации синтетической рабочей нагрузки для тестирования облачного приложения.
26. Характеристики рабочей нагрузки облачного приложения. Показатели производительности для облачных приложений.
27. Типы тестирований облачных приложений.

## 2. КРИТЕРИИ ОЦЕНКИ ПО ДИСЦИПЛИНЕ

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине				
Оценка	2 (не зачтено)	3 (зачтено)	4 (зачтено)	5 (зачтено)
виды оценочных средств				
<b>Знания</b> (виды оценочных средств: приведены в п. 1.2.)	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
<b>Умения</b> (виды оценочных средств: приведены в п. 1.2.)	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
<b>Навыки</b> (владения, опыт деятельности) (виды оценочных средств: приведены в п. 1.2..)	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