МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ, МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВО Кыргызско-Российский Славянский университет
имени первого Президента Российской Федерации Б.Н. Ельцина

УТВЕРЖДАЮ
и.о. декана факультета ПИФ
_____ Н. М. Комаров
22.09.2025

Проектирование высоконагруженных систем

рабочая программа дисциплины (модуля)

Закреплена за кафедрой **Информационных и вычислительных технологий**

Учебный план g090404_25_1пи_рпис.plx
Направление подготовки 09.04.04 - РФ, 710400 - КР Программная инженерия
Магистерская программа "Разработка программно-информационных систем"

Квалификация **магистр**

Форма обучения **очная**

Общая трудоемкость **4 ЗЕТ**

Часов по учебному плану 144
в том числе:
аудиторные занятия 38
самостоятельная работа 105,9

Виды контроля в семестрах:
зачет с оценкой 3

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	**3 (2.1)**		Итого	
Недель	18			
Вид занятий	УП	РП	УП	РП
Лекции	16	16	16	16
Практические	22	22	22	22
Контактная работа в период теоретического обучения	0,1	0,1	0,1	0,1
В том числе инт.	8	8	8	8
В том числе в форме практ.подготовки	22	22	22	22
Итого ауд.	38	38	38	38
Контактная работа	38,1	38,1	38,1	38,1
Сам. работа	105,9	105,9	105,9	105,9
Итого	144	144	144	144

Программу составил(и):
ст. преп., Беляев А.А.;ст. преп., Гайдамако В.В. _____

Рецензент(ы):
к.т.н., доцент , Демиденко А.П. _____

Рабочая программа дисциплины
Проектирование высоконагруженных систем

разработана в соответствии с ФГОС 3++:
Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.04 Программная инженерия (приказ Минобрнауки России от 19.09.2017 г. № 932)

составлена на основании учебного плана:
Направление подготовки 09.04.04 - РФ, 710400 - КР Программная инженерия
Магистерская программа "Разработка программно-информационных систем"
утвержденного учёным советом вуза от 30.06.2025 протокол № 13

Рабочая программа одобрена на заседании кафедры
Информационных и вычислительных технологий

Протокол от 03.09.2025 г. № 1
Срок действия программы: 2025-2026 уч.г.
Зав. кафедрой д.т.н., проф. Лыченко Н.М.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ
к рабочей программе дисциплины (модуля)

Проектирование высоконагруженных систем

наименование дисциплины

№ п/п	Основание внесения изменения	Раздел/ пункт	Содержание изменения
1.	№357-П от 24 июня 2026- О преобразовании Естественно-технического факультета и Факультета архитектуры дизайна и строительства в Политехнический институт КРСУ	Титульный лист. Пункт Разработано	Заменить слова «Разработано Кафедра Информационных и вычислительных технологий» словами «Разработано: Высшая школа Цифровых технологий»
2.			
3.			
4.			
5.			

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Сформировать у обучаемого навыки проектирования высоконагруженных программных систем;
1.2	Сформировать представление об основных понятиях, принципах и подходах, используемых при проектировании высоконагруженных систем;
1.3	Сформировать представление о методах тестирования и обеспечения качества высоконагруженных систем.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП	
Цикл (раздел) ООП:	Б1.В.ДВ.01
2.1	**Требования к предварительной подготовке обучающегося:**
2.1.1	Проектирование распределенных информационных систем
2.1.2	Системы обработки экспериментальных данных
2.1.3	Производственная практика по получению профессиональных умений и опыта профессиональной деятельности
2.1.4	Теория вычислительных процессов
2.1.5	Принципы WEB - программирования
2.2	**Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:**
2.2.1	Преддипломная практика
2.2.2	Научно-исследовательская работа
2.2.3	Выполнение и защита выпускной квалификационной работы

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ПК-1: Владение навыками программной реализации систем с параллельной обработкой данных и высокопроизводительных систем	
Знать:	
Уровень 1	методы проектирования систем с параллельной обработкой данных, высокопроизводительных систем и их компонент; основные элементы архитектуры высокопроизводительных вычислительных систем.
Уметь:	
Уровень 1	создавать алгоритмы параллельных вычислений; организовать взаимодействие процессов(потоков) и синхронизацию доступа к разделяемым ресурсам с помощью современных методов.
Владеть:	
Уровень 1	средствами параллельного программирования; способами проверки правильности работы программных комплексов с параллельной обработкой данных.

В результате освоения дисциплины обучающийся должен

3.1	**Знать:**
3.1.1	- основные характеристики, описывающие производительность, надежность, доступность системы, их связь с нагрузкой;
3.1.2	- понятие высокой нагрузки, единицы измерения нагрузки;
3.1.3	- архитектуру и подходы к разработке высоконагруженных систем
3.1.4	- основные причины потери производительности системы при повышении нагрузки;
3.1.5	- основные методы повышения производительности сетевой подсистемы;
3.1.6	- методы масштабирования на всех уровнях трехзвенной архитектуры;
3.1.7	- методы тестирования высоконагруженных систем.
3.2	**Уметь:**
3.2.1	- определять характеристики производительности системы;
3.2.2	- анализировать требования к системе, связанные с нагрузкой на систему;
3.2.3	- планировать процесс разработки высокопроизводительных систем;
3.2.4	- проектировать системы с использованием оптимальных для обеспечения производительности приемов;
3.2.5	- оптимизировать системы с повышенными требованиями к производительности;
3.2.6	- взаимодействовать с командой тестирования при выполнении нагрузочных тестов.
3.3	**Владеть:**

3.3.1	- современными методами и подходами к проектированию высоконагруженных программных систем на примере WEB-приложений;
3.3.2	- методами оценки производительности системы;
3.3.3	- методами оптимизации производительности высоконагруженных систем;
3.3.4	- навыками организации тестирования приложений при больших нагрузках.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетен-ции	Литература	Инте ракт.	Пр. подг.	Примечание
	Раздел 1. Введение. Сетевая подсистема							
1.1	Демонстрация сервера и клиента , разъяснение принципов взаимодействия процессов с использованием сокетов на С. /Пр/	3	2	ПК-1	Л1.1 Л1.2Л2.2 Л2.3 Э1 Э2	2	2	Обсуждение реализации сокетов
1.2	Проработка лекционного материала, изучение кода демонстрационного клиент-серверного приложения. /Ср/	3	9,9	ПК-1	Л1.1 Э1 Э2			
1.3	Понятие высокой нагрузки, высокой надежности, высокой производительностям, высокой доступности. Примеры систем с высокой нагрузкой. Высокая нагрузка для Интернет-приложений. Архитектура WEB-сервера. Синхронный и асинхронный сервер. Единицы измерения нагрузки. Баланс между удобством разработки и производительностью. /Лек/	3	2	ПК-1	Л1.1Л2.1 Э1 Э2	1		Интерактивное обсуждение примеров систем с высокой нагрузкой
1.4	HTTP протокол и WEB-оптимизация Оптимизация времени загрузки страницы в различных браузерах. Обзор возможностей HTTP и особенностей браузеров используемых для оптимизации времени загрузки страницы. Keep-alive, pipelining, deflate, chunked, несколько соединений на домен, cookies /Лек/	3	4	ПК-1	Л1.1Л2.1 Э1 Э2	1		Интерактивное обсуждение взаимодействия процессов в сети
1.5	Практическая работа №1 Разработка базового http сервера на языке по выбору /Пр/	3	4	ПК-1	Л1.1 Э1 Э2		4	
1.6	Проработка лекционного материала, Выполнение практической работы №1 /Ср/	3	8	ПК-1	Л1.1 Э1 Э2			
1.7	Влияние сетевой подсистемы на производительность. Ограничения и проблемы сетевой подсистемы. Задержки бэкендов и клиентов, влияние на производительность и методы борьбы с ними. Подстройка TCP/IP стека под высокую нагрузку. /Лек/	3	4	ПК-1	Л1.1Л2.1 Э1 Э2			

	Раздел 2. Влияние аппаратного обеспечения на производительность							
2.1	Оперативная память Организация физической памяти, кеширование, Влияние на производительность. Обзор практических приемов работы с памятью позволяющих увеличить производительность. Настройка производительности /Лек/	3	2	ПК-1	Л1.1 Э1 Э2			
2.2	Практическая работа №2. Управление памятью в ОС Linux/Windows. Мониторинг производительности /Пр/	3	4	ПК-1	Л1.1 Э1 Э2		4	
2.3	Проработка лекционного материала, Выполнение практической работы №2 /Ср/	3	12	ПК-1	Л1.1Л2.1 Э1 Э2			
	Раздел 3. Масштабирование							
3.1	Масштабирование архитектурного решения. Вертикальное и горизонтальное масштабирование. Трехзвенная архитектура. Масштабирование фронтенда. Шардинг, разделение быстрых и медленных запросов. Кэширование. Методы балансировки сетевой нагрузки. Методы резервирования нагрузки. Отказоустойчивость фронтенда. Масштабирование бэкенда. Функциональное разделение. Классическое горизонтальное масштабирование. Масштабирование во времени. Очереди. Управление конфигурацией и мониторинг. /Лек/	3	2	ПК-1	Л1.1Л2.1 Э1 Э2	1		Какие структуры данных используются в NoSQL базах данных?
3.2	Практическая работа №3 Разработка трехзвенного WEB приложения на базе HTTP сервера /Пр/	3	2	ПК-1	Л1.1 Э1 Э2		2	
3.3	Проработка лекционного материала, Выполнение практической работы №3 /Ср/	3	12	ПК-1	Л1.1Л2.1 Э1 Э2			
	Раздел 4. Системы хранения							

4.1	Базы данных и дисковая подсистема. Исследование предметной области, исследование потоков данных. Типы баз данных.Big Data. Выбор СУБД Производительность линейного, случайного и конкурентного доступа. Настройка производительности /Лек/	3	2	ПК-1	Л1.1 Э1 Э2	1		Интерактивное обсуждение методов повышения производительности БД
4.2	Практическая работа №4. Настройка производительности БД – разработка логической и физической БД /Пр/	3	2	ПК-1	Л1.1 Э1 Э2		2	
4.3	Проработка лекционного материала. Выполнение практической работы №4 /Ср/	3	12	ПК-1	Л1.1Л2.1 Э1 Э2			
4.4	Практическая работа №4. Настройка производительности – структура запросов, план запроса, запросы с join в распределенных серверах /Пр/	3	2	ПК-1	Л1.1Л2.1 Э1 Э2		2	
4.5	Самостоятельное изучение способов повышения производительности БД, методов тестирования производительности БД. Выполнение практической работы №4 /Ср/	3	12	ПК-1	Л1.1 Э1 Э2			
4.6	Проработка лекционного материала, изучение кода демонстрационного клиент-серверного приложения. /Ср/	3	12	ПК-1	Л1.1Л2.1 Э1 Э2			
4.7	Практическая работа №4. Тестирование производительности приложения /Пр/	3	2	ПК-1	Л1.1 Э1 Э2		2	
4.8	Проработка литературы, публикаций в сети Интернет, подготовка реферата, Выполнение практической работы №4 /Ср/	3	16	ПК-1	Л1.1 Э1 Э2			
4.9	Защита рефератов /Пр/	3	2	ПК-1	Л1.1 Э1 Э2	1	2	Интерактивное обсуждение рефератов
4.10	Выполнение практической работы №4 – работа над итоговым проектом /Ср/	3	12	ПК-1	Л1.1Л2.1 Э1 Э2			
4.11	Защита итогового проекта /Пр/	3	2	ПК-1	Л1.1 Э1 Э2	1	2	Интерактивное обсуждение проектов
4.12	/КрТО/	3	0,1	ПК-1				

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

Контрольные вопросы для проверки уровня обученности ЗНАТЬ:

1. Что такое высоконагруженная система? Приведите примеры систем с высокой нагрузкой
2. Расскажите о традиционных качествах высоконагруженных систем(растущее количество пользователей, растущее количество данных, изменения паттернов поведения пользователей0

3. Что такое производительная система?
4. Особенности управления ресурсами в высоконагруженных системах
5. Особенности разработки высоконагруженных систем
6. Почему высоконагруженные системы – в большинстве случаев являются Интернет-приложениями
7. Трехзвенная архитектура высоконагруженных Интернет приложений
8. Специализированные группы серверов по типам нагрузки
9. Для чего нужно масштабирование? Примеры
10. Что такое Slashdot-эффект, habrahabr-эффект?
11. Что такое DDOS атака?
12. WEB-приложения - структура типового сайта, LAMP-технология
13. Назовите известные Вам способы подключения динамического содержимого, кратко охарактеризуйте их.
14. Недостатки и достоинства CGI(FastCGI, UWSGI). Почему CGI не используется в высоконагруженных системах?
15. Недостатки и достоинства FastCGI. Почему CGI технологии не используются в высоконагруженных системах?
16. Недостатки и UWSGI. Почему CGI технологии не используются в высоконагруженных системах?
17. Недостатки и достоинства mod_php. Способы повышения эффективности
18. Недостатки и достоинства mod_perl. Способы повышения эффективности
19. Недостатки и достоинства java.
20. Недостатки и достоинства node.js
21. Недостатки и достоинства javascript
22. Недостатки и достоинства использования LUA для web-серверов
23. Компилируемые и динамические (интерпретируемые) языки. Компиляция Just-in-time
24. Архитектура web-сервера, классическая и асинхронная обработка. Архитектура apache/apache2/thttpd/nginx.
25. Архитектура web-сервера, классическая и асинхронная обработка. Архитектура thttpd/nginx.
26. Какие сервера – классические или с асинхронной обработкой предпочтительнее использовать для высоконагруженных систем? Почему?
27. Медленные клиенты. Reverse-proxy для мобильных клиентов
28. Значение мониторинга для сопровождения и развития высоконагруженных приложений
29. Задачи сбора и обработки статистики,
30. Задачи разработки приложения в контексте высокой нагрузки
31. Что такое доступность?
32. Что такое надежность? Основные приемы повышения надежности
33. Какие характеристики производительности системы Вы знаете?
34. Как нагрузка влияет на производительность?
35. Каковы причины потери производительности?
36. Приведите примеры известных Вам высоконагруженных систем
37. Как измеряется нагрузка? Единицы измерения нагрузки
38. Как повысить производительность при увеличении нагрузки, методы
39. Особенности тестирования высоконагруженных систем
40. Расскажите о трехзвенной архитектуре
41. Расскажите о группах серверов по типам нагрузки
42. Что такое шардинг?
43. Что такое горизонтальное масштабирование?
44. Что такое вертикальное масштабирование?
45. Влияние сетевой подсистемы на производительность – сетевые задержки, скорость света и расстояние между ДЦ.
46. Опишите принцип работы и особенности HTTP.
47. Преимущества и недостатки реляционных баз данных
48. Преимущества и недостатки нереляционных (пост-реляционных) баз данных
49. Методы повышения производительности запросов к реляционным базам данных
50. Методы повышения производительности запросов к распределенным базам данных. Запросы с join
51. Методы ускорения систем построенных на базах данных: тюнинг, репликация.
52. Шардинг серверов баз данных.
53. Минимизация сетевой задержки при обращении к базам данных, , написание специализированной БД.
54. Когда базу данных можно отнести с Big Data?
55. Что такое «набор V» в Big Data?
56. Какие технологии обработки данных используются в Big Data?
57. Особенности настройки производительности для OLAP и OLTP баз данных

Контрольные задачи для проверки уровня обученности УМЕТЬ и ВЛАДЕТЬ:

1. TCP протокол: handshake, packetloss и TCP-retransmit
2. Задержки бэкендов и клиентов(фронтендов), влияние на производительность и методы борьбы.
3. Сетевые проблемы и различные рецепты (UDP, multicast, Jumbo-frames, socket на процесс, многопоточные сетевые карты).
4. Подстройка TCP/IP стека под высокую нагрузку.
5. Назовите коды статусов HTTP ответа предназначенных для redirect.
6. Опишите работу блокирующегося TCP/IP сервера. Назовите достоинства и недостатки
7. Опишите работу TCP/IP сервера с неблокирующей обработкой. Назовите достоинства и недостатки
8. Опишите работу TCP/IP сервера с неблокирующей обработкой. Какие библиотеки могут использоваться для

создания таких серверов? Примеры серверов с неблокирующей обработкой
9. Назовите принцип работы cookie.
10. Назовите принцип ограничения cookie.
11. Что такое same origin policy?
12. Назовите принцип работы session
13. Какие framework для разработки сайтов вы знаете?
14. Назовите коды статусов HTTP ответа предназначенных для redirect.
15. Что такое AJAX? Для решения каких задач применяется AJAX?
16. Какие недостатки AJAX вы знаете?
17. Возможна ли загрузка файлов на сервер с использованием AJAX?
18. Что такое XML? Приведите пример задачи, в которой уместно использование XML.
19. Что такое JSON? Приведите пример задачи, в которой уместно использование JSON.
20. Что такое web-сервисы?
21. Что такое XML-RPC? Приведите пример задачи, в которой уместно использование XML-RPC.
22. Что такое JSON-RPC? Приведите пример задачи, в которой уместно использование JSON-RPC
23. Что такое SOAP? Приведите пример задачи, в которой уместно использование SOAP.
24. Что такое WSDL?
25. Назовите методы и практики отладки web-сервисов.
26. Назовите основные методологии разработки.
27. Назовите основные этапы методологии Scrum.
28. Назовите основные этапы методологии Kanban.
29. Назовите основные этапы методологии XP.
30. Назовите инструменты для контроля версионности программного кода.
31. Назовите основные этапы непрерывной интеграции.
32. Назовите инструменты для непрерывной интеграции.
33. Назовите инструменты для автоматической сборки проектов.
34. Назовите инструменты для инспекции программного кода.
35. Назовите задачи, решаемые с помощью unit-тестирования.
36. Назовите инструменты для unit-тестирования.
37. Назовите задачи, решаемые с помощью функционального тестирования.
38. Назовите инструменты для функционального тестирования.
39. Назовите основные типы уязвимостей web-сайтов.
40. Назовите методы обнаружения уязвимости типа SQL-injection.
41. Назовите методы эксплуатации уязвимостей типа SQL-injection.
42. Назовите методы написания кода защищенного от уязвимости типа SQLinjection.
43. Назовите методы обнаружения уязвимости типа XSS.
44. Назовите методы эксплуатации уязвимостей типа XSS.
45. Назовите методы написания кода защищенного от уязвимости типа XSS.

5.2. Темы курсовых работ (проектов)

не предусмотрены

5.3. Фонд оценочных средств

Задания к практическим работам
Практическая работа №1. Разработка базового http сервера на языке по выбору и его тестирование с повышением нагрузки
1. Разработать простой (синхронный) http сервер, без обращения к базам данных, выдающий одну страницу, содержащую несколько изображений (3-6) в ответ на запрос (страницу можно скачать с какого-либо существующего сервера).
2. Разработать средства тестирования, имитирующие нагрузку.
3. Провести тестирование с постепенным повышением нагрузки.

Практическая работа №2. Управление памятью в ОС Linux/Windows. Мониторинг производительности базового http сервера
1. Провести мониторинг загруки процессора и использования памяти системы (сервера) при работе http-сервера с постепенным повышением нагрузки.
2. Рассмотреть способы замедления потери производительности при повышении нагрузки

Практическая работа №3. Разработка трехзвенного WEB приложения
1. Разработать простое WEB приложение (можно использовать любые удобные технологии, любой сервер, обосновать выбор), с использованием СУБД в качестве системы хранения данных.
2. Провести тестирование, обнаружить пути повышения производительности при увеличении нагрузки

Практическая работа №4. Работа над итоговым проектом с использованием СУБД в качестве системы хранения данных, оптимизация обращения к БД.
1. Доработать приложение из ПР№3 с учетом высокой нагрузки.
2. Провести тестирование и оптимизацию производительности.

Вопросы к контрольной работе
1. Какие единицы нагрузки Вы знаете?
2. В чём может считаться количество данных в единицу времени?

3. Какой вариант обработчика, входящего http запроса позволяет реализовать наиболее эффективно несложную логику без обращений к базам данных?
4. Какова доступность сайта, если в среднем 1 час в месяц он недоступен?
5. Назовите возможные причины резкого повышения нагрузки на сайт
6. Опишите структуру LAMP сайта. Какие узкие места наиболее вероятны для LAMP сайтов?
7. Опишите последовательность операций клиента и сервера для подключения с установкой соединения для классического блокирующегося (синхронного) сервера
8. Что такое высоконагруженная система? Приведите примеры систем с высокой нагрузкой
9. Расскажите о традиционных качествах высоконагруженных систем(растущее количество пользователей, растущее количество данных, изменения паттернов поведения пользователей0
10. Что такое производительная система?
11. Особенности управления ресурсами в высоконагруженных системах
12. Особенности разработки высоконагруженных систем
13. Почему высоконагруженные системы – в большинстве случаев являются Интернет-приложениями
14. Трехзвенная архитектура высоконагруженных Интернет приложений
15. Специализированные группы серверов по типам нагрузки
16. Для чего нужно масштабирование? Примеры
17. Что такое Slashdot-эффект, habrahabr-эффект?
18. Что такое DDOS атака?
19. WEB-приложения - структура типового сайта, LAMP-технология
20. Назовите известные Вам способы подключения динамического содержимого, кратко охарактеризуйте их.
21. Недостатки и достоинства CGI(FastCGI, UWSGI). Почему CGI не используется в высоконагруженных системах?
22. Недостатки и достоинства FastCGI. Почему CGI технологии не используются в высоконагруженных системах?
23. Недостатки и UWSGI. Почему CGI технологии не используются в высоконагруженных системах?
24. Недостатки и достоинства mod_php. Способы повышения эффективности
25. Недостатки и достоинства mod_perl. Способы повышения эффективности
26. Недостатки и достоинства java.
27. Недостатки и достоинства node.js
28. Недостатки и достоинства javascript
29. Недостатки и достоинства использования LUA для web-серверов
30. Компилируемые и динамические (интерпретируемые) языки. Компиляция Just-in-time
31. Архитектура web-сервера, классическая и асинхронная обработка. Архитектура apache/apache2/thttpd/nginx.
32. Архитектура web-сервера, классическая и асинхронная обработка. Архитектура thttpd/nginx.
33. Какие сервера – классические или с асинхронной обработкой предпочтительнее использовать для высоконагруженных систем? Почему?
34. Медленные клиенты. Reverse-proxy для мобильных клиентов
35. Значение мониторинга для сопровождения и развития высоконагруженных приложений
36. Задачи сбора и обработки статистики,
37. Задачи разработки приложения в контексте высокой нагрузки
38. Что такое доступность?
39. Что такое надежность? Основные приемы повышения надежности
40. Какие характеристики производительности системы Вы знаете?
41. Как нагрузка влияет на производительность?
42. Каковы причины потери производительности?
43. Приведите примеры известных Вам высоконагруженных систем
44. Как измеряется нагрузка? Единицы измерения нагрузки
45. Как повысить производительность при увеличении нагрузки, методы
46. Особенности тестирования высоконагруженных систем
47. Расскажите о трехзвенной архитектуре
48. Расскажите о группах серверов по типам нагрузки
49. Что такое шардинг?
50. Что такое горизонтальное масштабирование?
51. Что такое вертикальное масштабирование?
52. Влияние сетевой подсистемы на производительность – сетевые задержки, скорость света и расстояние между ДЦ.
53. Опишите принцип работы и особенности HTTP.

Темы рефератов
1. Архитектура и методика проектирования поисковых систем
2. Архитектура и методика проектирования Facebook
3. Архитектура и методика проектирования Amazon
4. Архитектура и методика проектирования Google
5. Архитектура и методика проектирования Yandex
6. Методика проектирования Google+
7. Методика проектирования ВКонтакте – и др.
8. Особенности REST API для высокопроизводительных систем
9. Выбор языка программирования для создания высоконагруженных приложений
10. Nginx. Архитектура, настройка

11. Node.js. Архитектура, настройка
12. СУБД. Обзор, примеры. Реляционные и нереляционные СУБД. NoSQL.
13. Мониторинг производительности. Сбор и анализ статистики.
14. Альтернативные технологии отрисовки контента: SVG, Canvas, WebGL. Анимация
15. Ajax
16. Методологии разработки (Scrum, Kanban, XP)
17. Методологии непрерывной интеграции
18. Методологии автоматизации администрирования, сборки и тестирования проектов
19. Безопасность web-приложений. Основные уязвимости

5.4. Перечень видов оценочных средств

Практическая работа; Контрольная работа; Реферат.
Виды шкал оценивания представлены в Приложении 1.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год
Л1.1	В.В. Коваленко	Проектирование информационных систем: Учебное пособие: Основная	М.: Форум: НИЦ ИНФРА 2014
Л1.2	Мациевский Н.С.	Реактивные веб-сайты. Клиентская оптимизация в алгоритмахи примерах: [Электронный ресурс] : учебное пособие	Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование 2017

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год
Л2.1	А.В.Затонский	Информационные технологии: разработка информационных моделей и систем: Учеб. пос.: Дополнительная	М.:ИЦ РИОР: НИЦ ИНФРА 2014
Л2.2	О. Бунин и др	По материалам конференции Highload. Разработка высоконагруженных систем.	М: Из-во Олега Бунина 2012
Л2.3	Рудофф Эндрю, Стивенс Уильям Ричард, Феннер Билл	UNIX разработка сетевых приложений.	СПб: Питер 2007

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Центр разработки Windows Azure	http://msdn.microsoft.com/windowsazure/
Э2	Pattern Windows Azure Design Pattern Catalog	http://neudesic.blob.core.windows.net/azuredesignpatt

6.3. Перечень информационных и образовательных технологий

6.3.1 Компетентностно-ориентированные образовательные технологии

6.3.1.1	Изучение дисциплины студентами осуществляется в форме лекций, практических занятий в аудиторных условиях (лекционные аудитории и компьютерные классы), выполнения заданий на самостоятельную работу, контроля знаний.
6.3.1.2	При проведении лекций используются интерактивные формы обучения.
6.3.1.3	Практические занятия проводятся в компьютерных классах, оснащенных персональными компьютерами с необходимыми параметрами и с установленным необходимым программным обеспечением. Используется Интернет для получения дополнительной информации. Для интерактивной формы обучения используются защита работ перед группой, презентация.
6.3.1.4	Экзамен проводится в письменной форме в виде ответов на вопросы.

6.3.2 Перечень информационных справочных систем и программного обеспечения

6.3.2.1	Операционные системы: Linux, Windows.
6.3.2.2	СУБД: MS SQL, Windows Azure, Hadoop.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1	Учебная лаборатория компьютерного моделирования и информационных технологий -
7.2	ПК- 13 шт;
7.3	сервер -1; ПК-преподавателя-1;
7.4	локальная сеть кафедры;
7.5	интернет со скоростью 70 Мбит/сек.;

7.6	зона WI-FI.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Технологическая карта дисциплины представлена в Приложении №2.
Цель практических занятий – приобретение навыков разработки, тестирования, настройки производительности WEB-приложения с использованием базы данных в качестве системы хранения
Описание практических работ представлено в электронной папке преподавателя в локальной сети кафедры ИВТ.