

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»**

А.А. ФИЛИППОВ

**РАСПРЕДЕЛЕННЫЕ И ПАРАЛЛЕЛЬНЫЕ
ВЫЧИСЛЕНИЯ**
практикум по дисциплине
«Распределенные вычисления и приложения»

Ульяновск

УлГТУ

2019

Рекомендовано научно-методической комиссией факультета информационных систем и технологий в качестве практикума.

Филиппов, Алексей Александрович

Распределенные и параллельные вычисления : практикум / А. А. Филиппов. – Ульяновск : УлГТУ, 2019. – 13 с.

Практикум адресован студентам для выполнения и оформления лабораторных работ по дисциплине «Распределенные вычисления и приложения». Предоставлены варианты заданий, рекомендации и требования к лабораторным работам, разработанные в соответствии с рабочей программой дисциплины. Предназначен для студентов, обучающихся по направлениям: 09.03.03 «Прикладная информатика (профиль Прикладная информатика в экономике)», 09.03.04 «Программная инженерия (профиль Программная инженерия)».

Работа подготовлена на кафедре «Информационные системы».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Собеседование по лабораторным работам	5
Экзамен	5
Лабораторная работа №1	7
Контрольные вопросы к лабораторной работе	7
Лабораторная работа №2	8
Контрольные вопросы к лабораторной работе	8
Лабораторная работа №3	9
Контрольные вопросы к лабораторной работе	9
Лабораторная работа №4	10
Контрольные вопросы к лабораторной работе	10
Варианты заданий	11
Список используемой литературы	12

ВВЕДЕНИЕ

Целью освоения дисциплины «Распределенные вычисления и приложения» является формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний о принципах разработки параллельных и распределенных приложений. Также рассматриваются особенности, типы и модели облачных вычислений. Особое внимание уделяется изучению подходов к проектированию и разработке параллельных и распределенных приложений с применением языка программирования Java, фреймворка Spring Boot и платформы Apache Ignite.

Задачами дисциплины являются:

- изучение принципов функционирования параллельных и распределенных приложений;
- формирование навыков проектирования параллельных и распределенных приложений;
- рассмотрение и изучение применения различных подходов к построению параллельных и распределенных приложений;
- приобретение теоретических знаний и практических навыков разработки параллельных и распределенных систем.

Кроме того, в результате изучения дисциплины «Распределенные вычисления и приложения» обучающиеся на основе приобретенных знаний, умений и навыков достигают освоения компетенций на определенном уровне их формирования.

Тематический план дисциплины:

1. Распределенные вычисления и приложения

- 1.1. Понятие распределенных вычислений и распределенной системы.
- 1.2. Цели построения распределенных систем.
- 1.3. Требования к распределенным системам.
- 1.4. Взаимодействие в распределенных системах.
- 1.5. Модель распределенных систем.
- 1.6. Причинно-следственный порядок событий.
- 1.7. Взаимное исключение в распределенных системах.
- 1.8. Централизованные и распределенные алгоритмы.
- 1.9. Облачные вычисления.
- 1.10. Программное обеспечение как услуга (SaaS).
- 1.11. Платформа как услуга (PaaS).
- 1.12. Инфраструктура как услуга. (IaaS).

2. Реализация параллельных и распределенных приложений

- 2.1. Введение в Java Concurrency.
- 2.2. Java Concurrent Collections.
- 2.3. Java Queues.

- 2.4. Java Synchronizers.
- 2.5. Java Executors.
- 2.6. Java Locks.
- 2.7. Java Atomics.
- 2.8. Введение в MPI.
- 2.9. Введение в Spring Boot.
- 2.10. Введение в Apache Ignite.
- 2.11. Архитектура Apache Ignite.
- 2.12. Распределенные вычисления в Apache Ignite.

Собеседование по лабораторным работам

Собеседование по выполнению лабораторных работ осуществляется с целью проверки уровня знаний, умений, владений, понимания студентом основных методов и методик проектирования, разработки и поддержки информационных систем в задачах автоматизации бизнес-процессов, умения применять на практике полученных знаний. Каждое лабораторное занятие студент выполняет объемную задачу по конкретной теме с возможностью внесения доработок и изменений. Шкала оценивания имеет вид (таблица 1)

Таблица 1

Шкала и критерии оценивания решения задач на лабораторных занятиях

Оценка	Критерии
Сдано	Студент демонстрирует знания теоретического и практического материала по теме лабораторной работы, дает правильный алгоритм решения, в конце занятия студент выдает законченную и полностью функционирующую разработку.
Не сдано	Студент в конце занятия выдает не законченную и/или не полностью функционирующую разработку, некорректно отвечает на дополнительные вопросы.

Экзамен

Экзамен по дисциплине проводится в форме решения на компьютере (написания программы) практических задач по билетам, а также ответа на сопутствующие теоретические вопросы. Билет содержит практическое задание (задачу) для контроля освоения умений и навыков всех запланированных в ходе изучения дисциплины компетенций. Билет формируется таким образом, чтобы в него попали практические задания, контролирующие уровень усвоения всех заявленных дисциплинарных компетенций.

Шкала оценивания имеет вид (таблица 2)

Таблица 2

Шкала и критерии оценивания экзамена

Оценка	Критерии
Отлично	Выставляется обучающемуся, если студент выполнил в полном объеме практическое задание и способен обосновать свои решения
Хорошо	Выставляется обучающемуся, если студент выполнил практическое задание не в полном объеме (не менее $\frac{3}{4}$) либо в полном объеме, но с некоторыми погрешностями и ошибками
Удовлетворительно	Выставляется обучающемуся, если студент выполнил практическое задание не в полном объеме (не менее $\frac{1}{2}$) либо в полном объеме, но с существенными погрешностями и ошибками
Неудовлетворительно	Выставляется обучающемуся, если студент не справился с выполнением практического задания

Лабораторная работа №1

Разработка многопоточного приложения с использованием Java Concurrency согласно варианту задания.

Необходимо:

1. Разработать однопоточный вариант алгоритма и замерить время его работы.
2. Разработать параллельный вариант алгоритма с использованием ThreadPoolExecutor и замерить время его работы.
3. Разработать параллельный вариант алгоритма с использованием ForkJoinPool и замерить время его работы.

Контрольные вопросы к лабораторной работе

1. В чем особенность разработки многопоточного приложения?
2. Что такое критические области?
3. Какие механизмы ОС используются для устранения проблемы попадания в критическую область?
4. Какие коллекции содержит пакет Java Concurrency?
5. Для чего могут быть использованы очереди из пакета Java Concurrency?
6. Для чего могут быть использованы синхронизаторы из пакета Java Concurrency?
7. Для чего могут быть использованы исполнители из пакета Java Concurrency?
8. Для чего могут быть использованы блокировщики из пакета Java Concurrency?
9. Для чего могут быть использованы атомарные типы данных из пакета Java Concurrency?

~~Лабораторная работа №2~~

~~Разработка параллельного MPI приложения на языке Java.~~

~~Необходимо разработать параллельный вариант алгоритма с применением MPI и
замерить время его работы.~~

~~Контрольные вопросы к лабораторной работе~~

- ~~1. Что такое MPI?~~
- ~~2. Для чего может быть использован MPI?~~
- ~~3. Какие реализации MPI существуют?~~
- ~~4. Каковы особенности работы с MPI в Java?~~

Лабораторная работа №3

Разработка распределенного приложения с использованием фреймворка Spring Boot.

Необходимо разработать параллельный вариант алгоритма с применением сервис-ориентированного подхода и фреймворка Spring Boot, замерить время его работы.

Контрольные вопросы к лабораторной работе

1. Каковы основные парадигмы фреймворка Spring Boot?
2. Какая подсистема фреймворка Spring Boot позволяет создать распределенные приложения?
3. Какие механизмы вызова удаленных процедур существуют?
4. В чем разница между синхронным и асинхронным вызовом удаленной процедуры?
5. Какие механизмы фреймворка Spring Boot позволяют использовать асинхронный вызов удаленных процедур?

Лабораторная работа №4

Разработка распределенного приложения с использованием платформы Apache Ignite.

Необходимо разработать параллельный вариант алгоритма с применением подхода Grid Gain и платформа Apache Ignite, замерить время его работы.

Контрольные вопросы к лабораторной работе

1. Для чего может быть использован Apache Ignite?
2. Каковы архитектурные особенности Apache Ignite?
3. В чем особенность настройки Apache Ignite?
4. Какие типы узлов могут быть в Apache Ignite?
5. Какие функции Apache Ignite позволяют выполнять распределенные вычисления?

Варианты заданий

1. Разделить элементы матрицы на наибольший элемент.
2. Разделить элементы матрицы на наименьший элемент.
3. Разделить элементы матрицы на среднее арифметическое всех ее элементов.
4. Определить минимальный элемент матрицы.
5. Определить максимальный элемент матрицы.
6. Определить среднее арифметическое элементов матрицы.
7. Определить минимальный элемент матрицы выше главной диагонали.
8. Определить максимальный элемент матрицы выше главной диагонали.
9. Определить среднее арифметическое элементов матрицы выше главной диагонали.
10. Определить минимальный элемент матрицы ниже главной диагонали.
11. Определить максимальный элемент матрицы ниже главной диагонали.
12. Определить среднее арифметическое элементов матрицы ниже главной диагонали.
13. Определить сумму из произведений элементов каждой строки матрицы.
14. Определить сумму из произведений элементов каждого столбца матрицы.
15. Упорядочить столбцы матрицы по убыванию суммы их элементов.
16. Упорядочить столбцы матрицы по возрастанию суммы их элементов.
17. Упорядочить строки матрицы по убыванию суммы их элементов.
18. Упорядочить строки матрицы по возрастанию суммы их элементов.
19. Упорядочить строки матрицы по убыванию первых элементов.
20. Упорядочить строки матрицы по возрастанию первых элементов.
21. Упорядочить столбцы матрицы по убыванию первых элементов.
22. Упорядочить столбцы матрицы по возрастанию первых элементов.
23. Определить сумму элементов матрицы.
24. Упорядочить строки матрицы по возрастанию наименьших элементов.
25. Упорядочить строки матрицы по убыванию наибольших элементов.
26. Упорядочить столбцы матрицы по возрастанию наименьших элементов.
27. Упорядочить столбцы матрицы по убыванию наибольших элементов.

Список используемой литературы

1. Миков, А.И. Распределенные системы и алгоритмы [Электронный ресурс] : учебное пособие / А.И. Миков, Е.Б. Замятина. — Электрон. дан. — Москва : , 2016. — 246 с.
<https://e.lanbook.com/book/100446>
2. Биллиг, В.А. Параллельные вычисления и многопоточное программирование [Электронный ресурс] : учебное пособие / В.А. Биллиг. — Электрон. дан. — Москва : , 2016. — 310 с.
<https://e.lanbook.com/book/100361>
3. Свистунов, А.Н. Построение распределенных систем на Java [Электронный ресурс] : учебное пособие / А.Н. Свистунов. — Электрон. дан. — Москва : , 2016. — 317 с.
<https://e.lanbook.com/book/100371>
4. Антонов, А.С. Параллельное программирование с использованием технологии MPI [Электронный ресурс] : учебное пособие / А.С. Антонов. — Электрон. дан. — Москва : , 2016. — 83 с.
<https://e.lanbook.com/book/100359>
5. Левин, М.П. Параллельное программирование с использованием OpenMP [Электронный ресурс] : учебное пособие / М.П. Левин. — Электрон. дан. — Москва : , 2016. — 133 с.
<https://e.lanbook.com/book/100358>
6. Сердюк, Ю.П. Параллельное программирование для многоядерных процессоров [Электронный ресурс] : учебное пособие / Ю.П. Сердюк, А.В. Петров. — Электрон. дан. — Москва : , 2016. — 170 с.
<https://e.lanbook.com/book/100357>
7. Материалы сайта The Java Tutorials
<https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>
8. Материалы сайта MPJ Express
<http://mpj-express.org/guides.html>
9. Материалы сайта Spring Guides
<https://spring.io/guides/>
10. Материалы сайта Apache Ignite
<https://apacheignite.readme.io/docs/getting-started>

РАСПРЕДЕЛЕННЫЕ И ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ:

практикум к выполнению
лабораторных работ по дисциплине
«Распределенные вычисления и приложения»

Автор ФИЛИППОВ Алексей Александрович

УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.