Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Кафедра «Прикладные информационные технологии»

Методические указания к выполнению контрольной работы

по дисциплине
Б.1.3.11.1 "Основы грид - вычислений"

направления подготовки
09.03.02 "Информационные системы и технологии"
Профиль «Информационные системы и технологии»

Саратов 2019

1. Цель работы

Целью работы является формирование следующих компетенций:

- ОПК-6: способностью выбирать и оценивать способ реализации информационных систем и устройств (программно, аппаратно или программно-аппаратно) для решения поставленной задачи.
- ПК-11: способностью к проектированию базовых и прикладных информационных технологий.

Компетенции формируются путем разработки программного обеспечения позволяющего производить распределенные вычисления с использованием грид-системы OurGrid. В части компетенции ПК-11 цель достигается за счет проектирования и реализации комплексной информационной среды включающей самостоятельно разработанные и сторонние подсистемы и непосредственно пригодной к использованию. В части компетенции ОПК-6 цель достигается за счет проектирования алгоритма разбиения поставленной задачи на программные подзадачи и оценки эффективности принятого решения.

2. Порядок выполнения работы

В результате выполнения работы необходимо разработать систему, которая будет пригодна для выполнения грид вычислений в системе OurGrid по задаче выбранной студентом из списка, приведённого в перечне заданий. Итоговая система должна состоять из следующих элементов:

1. Программа, выполняющая задачи на вычислительном узле.
2. Программа, формирующая задачи, отправляющая их в грид среду и формирующая итоговый ответ из результатов вычислительных узлов.

Программное обеспечение для формирования задачи и их выполнения может быть выполнено студентом с использованием любой технологии, однако оно должно обеспечивать возможность быть запущенным кроссплатформенно с использованием грид среды.

Программное обеспечения для формирования заданий должно уметь генерировать задания в формате принимаемом системой OurGrid, дожидаться конца выполнения работы (либо уметь остановить выполнение работы при необходимости), и формировать ответ. Данное ПО должно иметь пользовательский интерфейс в виде окна либо веб-интерфейса. Посредством данного интерфейса пользовать должен уметь выставить настройки выполняемой Работы и увидеть ответ в приемлемом для задачи виде.

Программное обеспечение для вычислительного узла выполняется в форме консольного приложения, параметры которого устанавливаются при старте программы посредством параметров командой строки.

Таким образом, работа выполняется в следующем порядке:

1. Выбрать задание из списка в соответствии с двумя последними цифрами номера зачетной книжки (студенческого билета)
2. Разработать программное обеспечение
3. Продемонстрировать процесс выполнения исходной задачи с эмуляцией грид среды
4. Отчитаться по программному обеспечению

Работа засчитывается в том случае, если она удовлетворяет всем описанным требованиям, а также сдан отчет, который выполняется путем устного опроса студента по принципам решения представленной задачи и исходному коду разработанного программного обеспечения.

3. Варианты работы

№ Задачи	номер зачетной книжки									
1	1	12	23	34	45	56	67	78	89	00
2	2	13	24	35	46	57	68	79	90	
3	3	14	25	36	47	58	69	80	91	
4	4	15	26	37	48	59	70	81	92	
5	5	16	27	38	49	60	71	82	93	
6	6	17	28	39	50	61	72	83	94	
7	7	18	29	40	51	62	73	84	95	
8	8	19	30	41	52	63	74	85	96	
9	9	20	31	42	53	64	75	86	97	
10	10	21	32	43	54	65	76	87	98	
11	11	22	33	44	55	66	77	88	99	

4. Задания

1. Задача коммивояжёра

В качестве исходных данных задается матрица смежности графа, описывающая узлы графа (города) и вес ребер (стоимость дорог между городами. Необходимо найти дорогу с наименьшей суммой ребер, при этом проходящую через все города и возвращающуюся к исходному городу. Задача заключается в поиске самого выгодного маршрута, проходящего Маршрут может проходить каждый город не один раз, стартовый город может быть выбран произвольно.

2. Построение маршрута

В качестве исходных данных задается матрица смежности графа, описывающая узлы графа (города) и вес ребер (стоимость дорог между городами), и число N определяющее длину маршрута. Необходимо подобрать наиболее дешевый маршрут, имеющий длину N и не проходящий через один и тот же город дважды.

3. Расстановка ферзей

Необходимо на шахматной доске, размером N×N, расставить максимально возможное количество ферзей так, что бы они били каждую свободную клетку и не били друг друга. Математически это можно выразить как N×N, заполненная нулями и единицами таким образом, чтобы сумма всех элементов матрицы была равна N, при этом сумма элементов ни в одном столбце, строке или диагональном ряде матрицы не превышала единицы.

4. Подбор пароля

В качестве исходных данных выступает файл зашифрованный паролем длиной не более чем 20 символов формата ASCII. Необходимо подобрать пароль к этому файлу.

5. Таксисты и пассажиры

Таксомоторная компания имеет X свободных такси в разных точках города, и Y пассажиров. Необходимо подобрать такую комбинацию таксистов и пассажиров, что бы суммарная стоимость пути оказалась минимальна для такси. В качестве исходных данных задается матрица смежности графа, описывающая узлы графа (города) и вес ребер (стоимость дорог между городами, позиции таксистов, а также позиции пассажиров и их точки назначения.

6. Задача о ранце

Из заданного множества предметов со свойствами «стоимость», «вес» и «количество» требуется отобрать подмножество с максимальной полной

стоимостью, соблюдая при этом ограничение на суммарный вес не более N кг, где список и параметры предметов, а также N задается пользователем произвольно.

Предмет	Вес	Цена	Кол-во
A	1.5	1	5
B	2	1.5	8
C	5	3	4
D	7	6	3
E	8	6.5	4
F	8.5	8	3
G	9	10	2
H	10	15	1

Таблица 1. Пример характеристик предметов для ранца

7. Подматрица с наибольшей суммой

Дана матрица размером N×N, содержащая положительные и отрицательные числа. Необходимо найти квадратную подматрицу с максимально возможной суммой.

8. Поиск текста в книге

Пользователем задается два текста, необходимо посчитать сколько раз первый текст встречается внутри второго. Например, сколько раз “Болконский” встречается в романе “Война и мир”

9. Совпадения текстов

Дано два текста, необходимо выявить все отрывки первого текста встречающиеся во втором тексте. В расчёт принимаются только отрывки максимально возможно длины. Таким образом если в первом и втором тексте упоминается текст “мама мыла раму”, то слова “мама”, “мыла” и т.д. не учитываются.

10. Латинский квадрат N-го порядка

Латинский квадрат N-го порядка — таблица размеров N × N, заполненная N элементами множества M таким образом, что в каждой строке и в каждом столбце таблицы каждый элемент из M встречается в точности один раз. Пример латинского квадрата приведен в таблице 2.

A	C	B
B	A	C
C	B	A

Таблица 2. Пример латинского квадрата третьего порядка с буквами

Задача должна быть решена в условиях, когда часть ячеек таблицы заранее заполнена какими-либо значениями.

11.Поиск слов

Игра представляет собой поиск в таблице с буквами спрятанных слов по заданному словарю (см. рис1).

М	О	К	Й	А	Р	Т	И	О	К	Ь	Ы	С	Л
О	Р	Е	З	О	О	С	В	О	С	Л	С	Т	А
С	Д	Е	Г	Д	К	О	Н	О	Ч	А	Н	П	Н
Н	Е	А	И	А	Н	Л	М	Р	Л	У	И	О	О
А	Б	Т	Т	А	Е	Н	Р	К	А	В	Ж	Р	С
В	Н	М	З	Ч	А	Ц	Щ	Я	Ц	С	Д	О	Р
А	Д	Р	О	М	Р	Е	Г	Р	Ю	Б	П	Ш	Е
Ц	У	Г	Б	Р	Ж	С	Ч	В	С	Х	Р	А	П
Щ	Б	У	Р	Е	Б	Т	Э	Ж	З	Б	Ф	Ц	Д
С	Л	О	В	О	С	О	Ч	Е	Т	А	Н	И	Е
А	И	И	К	Т	Д	Я	Д	А	Е`	Р	М	Т	М
А	К	В	О	Р	И	Н	Е	Р	Т	Ы	Л	А	Я
А	А	Д	С	Я	Ю	И	Ы	Р	Ь	Н	Ь	Т	Х
Ж	Т	Ц	Ь	Д	Ш	Е	Д	Е	В	Я	Т	А	Я

АВАНСОМ
АНТИДОТ
БАГОР
БАРЫНЯ
БЕДРО
БЮРГЕР
ВКОСЬ
ВСХРАП
ВУАЛЬ
ДЕВЯТАЯ
ДЖИНСЫ
ДОБРОМ
ДУБЛИКАТ
ЕЖЕВИКА
ЗАМАХ
ЗАНОВО
КЛАСС
МЕДПЕРСОНАЛ
МОРДА
ОЗЕРО
ОТРЯД
ПОРОША
РАЙКОМ
РАСПАД
РОДНЫЕ
СЛОВОСОЧЕТАНИЕ
СОЛНЦЕСТОЯНИЕ
СОМНАМБУЛА
ТАКСИ
ТРЕНИРОВКА
ЦИТАТА
ЧЕЛНОК
ЧУЖОЕ

Рисунок 1. Пример поля со словами и словаря.

В данной работе необходимо реализовать поиск таким образом, что словарь содержит больше слов, чем находится в таблице (желательно взять полный словарь русского языка) и необходимо вычеркнуть из таблицы слова так, что бы минимизировать количество букв оставшихся в таблице после вычеркивания всех слов.

5. Литература

1. Oracle Java. Class String [Электронный ресурс] Режим доступа: https://docs.oracle.com/javase/7/docs/api/java/lang/String.html (Дата обращения 03.02.2018).
2. CodingBat code practice [Электронный ресурс] Режим доступа: http://codingbat.com (Дата обращения 01.02.2018).
3. Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала, том 1. 8-е издание.: Пер. с англ. - М.: "ООО И.Д. Вильямс", 2011. 816 с.
4. Online Java Compiler [Электронный ресурс] Режим доступа: https://www.jdoodle.com/online-java-compiler (Дата обращения 04.02.2018).

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Институт прикладных информационных технологий и коммуникаций
Направление «Информационные системы и технологии»
Кафедра «Прикладные информационные технологии»

КОНТРОЛЬНАЯ РАБОТА

по дисциплине «Основы грид вычислений»

Выполнил студент группы
ИФСТипуNN
заочной формы обучения
Иванов Игорь Владимирович
Номер зачетной книжки 111222
Проверил: доцент кафедры ПИТ
к.т.н. Ермаков А.В.

Саратов 2020

Целью работы является формирование компетенций:

ОПК-6: способностью выбирать и оценивать способ реализации информационных систем и устройств (программно, аппаратно или программно-аппаратно) для решения поставленной задачи.

ПК-11: способностью к проектированию базовых и прикладных информационных технологий.

Для формирования компетенции необходимо осуществить решение задачи разработки программного обеспечения выполнения заданий для грид-среды в соответствии с следующим заданием:

Это пример задания, вместо него необходимо вставить свое в соответствии с вариантом. Дана матрица чисел, необходимо посчитать сумму чисел этой матрицы.

Для данной задачи принято решение выполнять разбиение задачи на подзадачи путем выделения в качестве одной подзадачи одной строки матрицы. Таким образом первый вычислительный узел будет рассчитывать сумму чисел первой строки матрицы, второй узел – вторую и так далее, пока не будут получены суммы для каждой строки.

Таким образом, было сформировано два приложения:

Первое приложение формирует задания для грид среды и отправляет их на вычисление, а также ожидает получения файлов ответов от вычислительных узлов и складывает суммы строк для получения окончательного ответа.

Второе приложение будет использоваться вычислительным узлом для получения суммы одной строки.

Оба приложения разработаны с использованием языка программирования Java. Разберем исходный код приложений

Приложение для вычислительного узла располагается в классе Sum.

```
public class Sum {
    public static void main(String[] args) throws Exception {
        /*в качестве первого параметра командной строки ожидается номер строки в матрице, требующий суммирования на данном вычислительно узле*/
        int num=Integer.parseInt(args[0]);
/*в тестовых целях, название файла с матрицей данных задано жестко, однако ожидается что в реальном приложении оно будет указано относительно текущего места*/
        Scanner sc= new Scanner(new File("D:\\программирование\\JavaLessons\\Grid\\arr.txt"));
```

```
/*на вход приложения подается вся матрица целиком, поэтому мы пропускаем первые num строк, которые не относятся к текущей задаче*/
        for(int i=0;i<num;i++) sc.nextLine();
/*считываем искомую строку*/
        Scanner scanString= new Scanner(sc.nextLine());
        int sum=0;
/*проходимся по строке, разбиваем её на числа и складываем между собой*/
        for(int i=0;scanString.hasNextInt();i++){
           sum+=scanString.nextInt();
        }
//записываем файл результата
        File f= new File("D:\\программирование\\JavaLessons\\Grid\\out"+num+".txt");
        f.createNewFile();
        FileWriter fw= new FileWriter(f);
        fw.append(sum+"");
        fw.close();
    }
}
```

Приложение для формирования заданий и сбора результатов располагается в классе Grid.

```
public class Grid {
    public static void main(String[] args) throws Exception {
         //создаем файл задания
        File fileJDF=new File("myJob.jdf");
        fileJDF.createNewFile();
        FileWriter JdfWriter=new FileWriter(fileJDF);
```

```
//начинаем блок описания работы
        JdfWriter.append("job:\n");
        JdfWriter.append("\tname: sum\n");
```
/*каждая подзадача в качестве входных данных будет получать файл с названием arr.txt и программу которую необходимо запустить в виде файла Sum.class*/
```
        JdfWriter.append("\tinit:\n");
        JdfWriter.append("\t\tput arr.txt arr.txt\n");
        JdfWriter.append("\t\tput Sum.class Sum.class\n");
        Scanner sc=new Scanner(new File("arr.txt"));
        int taskCount;
```
/*считаем количество подзадач и записываем задачу вычислительного узла. Поскольку одной подзадачей является сложение одной строки то количество подзадач равно количеству строк. При этом запускаться подзадачи будут путем вызова java приложения описанного в классе grid.Sum с указанием номера строки матрицы для данной подзадачи*/
```
        for(taskCount=0;sc.hasNext();taskCount++){
            sc.nextLine();
            JdfWriter.append("task: remote: java grid.Sum "+taskCount+"\n");
```
/*в качестве результата каждой подзадачи будет получен файл out с номером соответствующим номеру строки*/
```
       JdfWriter.append("\t  final: get out"+taskCount+".txt out"+taskCount+".txt\n");
        }
        JdfWriter.close();
        //произошла и jdf запихнулся в грид, началось выполнение
        //ждем когда он нам вернет файлы результатов
        int sum=0;
        for(int i=0;i<taskCount;i++){
            File out=new File("out"+i+".txt");
            while(!out.exists()) Thread.sleep(100);
```

```
            Scanner scOut= new Scanner(out);
            sum+=scOut.nextInt();
        }
        System.out.println(sum);
    }
}
```