

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського» Інститут
прикладного системного аналізу
Кафедра системного проектування

ЗВІТ
про виконання лабораторної роботи №5 з дисципліни
«Паралельні обчислення»

Виконав:
Студент III курсу
Групи ДА-92
Шляхов Данило Сергійович
Варіант №23

Київ – 2022

- 1. Мета роботи:** Дослідження мережевих можливостей розподілених систем та реалізація клієнт-серверного додатку.
- 2. Склад робочого місця:**
 - **Обладнання:** IBM-сумісний персональний комп'ютер.
 - **Програмне забезпечення:** операційна система Windows, Java SDK версії 1.2.2 або вище.
- 3. Завдання:**

Розробити клієнт-серверний додаток, використовуючи Java RMI, Sockets або аналогічну технологію. Обов'язковим пунктом для отримання найвищої оцінки є реалізація одночасної підтримки багатьох клієнтів. Сервер має логувати роботу з клієнтами (який клієнт приєднався, яку дію цей клієнт виконує, який клієнт від'єднався).

Забороняється використання високорівневих фреймворків та технологій (наприклад, Spring) через достатньо просту реалізацію цієї лабораторної роботи. У випадку ж використання чогось подібного необхідно вивчити теоретичні відомості та роботу цієї технології на нижчих рівнях.
- 4. Завдання за варіантом 23:**

Система для знаходження максимального, мінімального значення, моди і медіани вектору в залежності від команди.

5. Хід роботи

Для виконання роботи була обрана мова програмування GoLang.

Для з'єднання клієнта та сервера протоколом TCP використовується пакет net.

Клієнт передає дані серверу в такому вигляді:

Призначення	Код команди	Розмір вектору	Вектор
Розмір	8 байт	8 байт	8 * розмір

Коди команд:

1. Знайти максимум

2. Знайти мінімум
3. Знайти медіану
4. Знайти моду

Сервер, отримавши пакет, оброблює його та повертає результат у вигляді, описаному нижче:

Призначення	Код відповіді	Результат
Розмір	8 байт	8 байт

Сервер повертає код 0, якщо успішно опрацював запит.

Результат – це `int64`, оскільки всі команди повертають лише одне число.

Розглянемо текст програми. Сервер починає слухати за портом 4545

```
listener, err := net.Listen("tcp", ":4545")
```

Приймає з'єднання з клієнтом.

```
con, err := listener.Accept()
```

Запускає паралельно обробку клієнта та починає очікувати наступного клієнта. За допомогою цього, сервер може одночасно обробляти декількох клієнтів.

```
go handler(con, clientCounter, &wg)
```

Обробка клієнта складається з очікування запиту:

```
_, err := clientReader.Read(request)
```

Після прочитання робиться перевірка, чи не від'єднався клієнт. Якщо ні – запит обробляється. Зчитується команда, розмір вектору, сам вектор:

```
command, vecsize, vector, err := ReadRequest(request)
```

Розраховується потрібне значення:

```
result, err := processRequest(command, vector)
```

Результат відправляється клієнту:

```
err = Respond(con, result)
```

Робота клієнта складається з з'єднання з сервером:

```
con, err := net.Dial("tcp", "127.0.0.1:4545")
```

Зчитується запит з консолі. Запит має вигляд «command, size of vector».

```
clientRequest, err := clientReader.ReadString('\n')
```

Запит обробляється, створюється вектор чисел з вказаним розміром, та заповнюється випадковими числами. Згодом записується як масив байтів та відправляється серверу.

```
comm, size := processRequest(clientRequest)  
_, err := con.Write(request)
```

Далі клієнт очікує на відповідь від серверу

```
n, err := serverReader.Read(serverResponse)
```

Перевіримо правильність роботи програми. Запустимо сервер:

```
2022/06/03 18:11:54 started listening...
```

Запустимо першого клієнта:

```
2022/06/03 18:12:19 accepted connection with client 0
```

Виконаємо запит на знайдення максимуму серед 5 чисел. Консоль клієнта:

```
Input in format `command, size of vector`  
1 - max, 2 - min, 3 - median, 4 - trend  
Input:  
1, 5  
[164 68 190 132 171]  
result = 190
```

Консоль серверу:

```
got request from 0 client, command = 1, vecsize = 5  
vector - [164 68 190 132 171]  
max - 190
```

Бачимо, що вхідні дані співпадають, результат виконання команди правильний, а клієнт отримав результат.

Нижче показані результати роботи решти команд. Знаходження мінімуму:

```
got request from 0 client, command = 2, vecsize = 5  
vector - [38 25 97 187 29]  
min - 25
```

Знаходження медіани:

```
got request from 0 client, command = 3, vecsize = 5  
vector - [136 29 134 82 103]  
median - 103
```

Знаходження моди:

```
got request from 0 client, command = 4, vecsize = 20  
vector - [152 174 124 111 182 125 20 35 66 80 69 66 157 149 130 74 39 177 86 74]  
trend - 66
```

Перевіримо, що сервер може оброблювати одночасно декількох клієнтів:

```
2022/06/03 18:20:19 accepted connection with client 1  
2022/06/03 18:20:23 accepted connection with client 2
```

Нижче відображена робота трьох клієнтів з сервером.

```
got request from 1 client, command = 2, vecsize = 5  
vector - [114 178 180 187 129]  
min - 114  
got request from 2 client, command = 3, vecsize = 7  
vector - [132 53 91 191 9 86 120]  
median - 91  
got request from 0 client, command = 1, vecsize = 5  
vector - [161 197 47 181 105]  
max - 197  
2022/06/03 18:20:57 client 1 closed connection by terminating process  
got request from 0 client, command = 1, vecsize = 5  
vector - [3 30 112 89 178]  
max - 178  
2022/06/03 18:21:04 client 0 closed connection by terminating process  
got request from 2 client, command = 3, vecsize = 5  
vector - [4 120 153 86 127]  
median - 120  
2022/06/03 18:21:08 client 2 closed connection by terminating process
```

Репозиторій – ЛАБА 5

Висновок

В ході виконання роботи був реалізований клієнт-серверний додаток, що може обробляти запити одночасно декількох клієнтів. Для цього обробка запитів виконується паралельно. Клієнти підключаються за протоколом TCP за допомогою пакету net. Додаток виводить на екран запити клієнтів, результат роботи, події пов'язані з підключенням клієнтів та їх відключенням.