

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Лабораторна робота №8

із дисципліни «Технології паралельних та розподілених обчислень»

на тему «Розробка алгоритмів для розподілених систем
клієнт-серверної архітектури»

Підготував:

Студент ФІОТ, гр. ІТ-83,

Троян О.С.

Перевірив:

пос. Дифучина О. Ю.

Київ 2020

Мета роботи: Розробити веб-застосування клієнт-серверної архітектури, що реалізує алгоритм множення матриць.

Завдання:

1. Розробити веб-застосування клієнт-серверної архітектури, що реалізує алгоритм множення матриць (або інший обчислювальний алгоритм, який був Вами реалізований іншими методами розподілених обчислень в рамках курсу «Паралельні та розподілені обчислення») на стороні сервера. Розгляньте два варіанти реалізації
 - 1.1. Дані для обчислень знаходяться на сервері
 - 1.2. Дані для обчислень знаходяться на клієнтській частині застосування.

60 балів.

2. Дослідити швидкість виконання запиту користувача при різних обсягах даних.

20 балів.

3. Порівняти реалізацію алгоритму в клієнт-серверній системи та в розподіленій системі з рівноправними процесорами.

20 балів.

Хід роботи

1. Клієнт був написаний на мові C#, за допомогою Asp.net core.
2. Роль веб сервера буде виконували локальний IIS.

При старті локального серверу клієнт отримує html розмітку для відображення (MVC), це дозволяє зручно надсилати дані.

В розмітці також присутній JavaScript за допомогою якого буду надсилатися CRUD запити, це дозволить максимально точно відправити лише потрібні дані, щоб запити не мали лишньої інформації.

Сервер в свою чергу буде перемножувати матриці і повертати результат.

1. Інтерфейс користувача

```
<tr>
  <td colspan="2" style="margin: auto;">
    <div id="matrix3"></div>
    <div id="operationTime" style="font-size: 35px;"></div>
  </td>
</tr>
<tr>
<tr>
  <td colspan="2">
    <table style="margin-left: auto; margin-right: auto;">
      <tr><th>Raw</th><td><input id="rawInt" type="number" value="5" /></td></tr>
      <tr><th>Column</th><td><input id="columnInt" type="number" value="5" /></td></tr>
      <tr>
        <th colspan="2">
          <input onclick="drawDefaultMatrix(false)" type="submit" value="Генерація матриць" />
        </th>
      </tr>
    </table>
  </td>
</tr>
```

Matrix A					Matrix B				
2	3	2	3	2	3	1	3	3	3
3	1	2	3	2	2	1	2	2	2
1	2	1	3	3	3	1	2	1	3
1	1	2	3	1	2	2	1	1	1
1	2	3	2	1	1	1	3	1	3
A * B									
26	15	25	19	27					
25	14	24	18	26					
19	13	21	14	22					
18	11	15	11	17					
21	11	18	13	21					
308644									
Raw					<input type="text" value="5"/>				
Column					<input type="text" value="5"/>				
<input type="button" value="Генерація матриць"/>									
<input type="button" value="Відправити значення"/>					<input type="button" value="Без значень"/>				

2. Паралельне множення матриць.

```

var tasks = new List<Task>();

for (var i = 0; i < item.FirstMatrix.Count; i++)
{
    item.FinalMatrix.Add(item: new List<int>());
    for (var j = 0; j < item.FirstMatrix[0].Count; j++)
    {
        item.FinalMatrix[i].Add(item: 0);
        tasks.Add(item: CalculateRowColumn(row: i, column: j));
    }
}

tasks.ForEach(t: Task => t.Start());
Task.WaitAll(tasks.ToArray());

```

Для множення використовується клас Task, 1 ітерація це 1 значення в результуючій таблиці.

3. Розрахунок, дані на стороні серверу.

В даному випадку ми надсилаємо GET запит з розмірністю матриць в параметрах запиту

```

function sendGetRequest() {
    var xhr = new XMLHttpRequest();
    var url = 'https://localhost:44379/api/Calculation/' + document.getElementById("rowInt").value + '/' + document.getElementById("columnInt").value;
    xhr.open("GET", url, true);
    xhr.setRequestHeader("Content-Type", "application/json");
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            var result = JSON.parse(xhr.responseText);
            drawMatrix("matrix1", result.firstMatrix);
            drawMatrix("matrix2", result.secondMatrix);
            drawMatrix("matrix3", result.finalMatrix);
            document.getElementById("operationTime").innerHTML = result.time;
            document.getElementById("operationTime").focus();
        }
    };
    xhr.send();
}

```

На стороні серверу API вуглядає так

```

[HttpGet(template: "{row:int}/{column:int}")]
Ссылка: 0
public IActionResult GetWithPara([FromRoute]int row, [FromRoute] int column)
{
    try
    {
        var item = new ThreeResultMatrix { FirstMatrix = _matrixCalculationService.GenerateMatrix(row, column) };
        var startTime:long = DateTime.Now.ToFileTimeUtc();
        _matrixCalculationService.MatrixMultiplication(item);
        item.Time = DateTime.Now.ToFileTimeUtc() - startTime;
        return new JsonResult(item);
    }
    catch (Exception ex)
    {
        return BadRequest(ex);
    }
}

```

Оскільки ми маємо розмірність то генеруємо матриці на стороні серверу. Часом на генерацію нехтуємо, адже за умовою дані на стороні серверу у нас вже є.

Далі розраховуємо добуток матриці і повертаємо його назад клієнту разом з розрахунком часу який було витрачено.

Результат:

Matrix A					Matrix B				
2	3	2	2	1	2	2	1	2	2
2	2	3	2	1	3	2	2	2	3
3	1	1	3	2	2	3	2	2	3
3	3	2	2	1	3	2	2	3	2
3	1	2	3	1	2	3	3	2	2
A * B									
25	23	19	22	25					
24	24	19	22	25					
24	23	19	23	22					
27	25	20	24	27					
24	23	18	23	23					
308734									
Row					<input type="text" value="5"/>				
Column					<input type="text" value="5"/>				
<input type="button" value="Генерація матриць"/>									
<input type="button" value="Відправити значення"/>					<input type="button" value="Без значень"/>				

4. Відправка матриць в виді даних відбувається за допомогою POST запиту в тіло якого вкладаються обидві матриці

```
function sendPostRequest(json) {
    var xhr = new XMLHttpRequest();
    var url = "https://localhost:44379/api/Calculation/";
    xhr.open("POST", url, true);
    xhr.setRequestHeader("Content-Type", "application/json");
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            var result = JSON.parse(xhr.responseText);
            drawMatrix("matrix3", result.finalMatrix);
            document.getElementById("operationTime").innerHTML = result.time;
            document.getElementById("operationTime").focus();
        }
    };
    xhr.send(json);
}
```

Сам контролер приймає значення і починає розрахунок.

```
[HttpPost]
Ссылка: 0
public IActionResult PostWithParams([FromBody]List<List<List<int>>> value)
{
    try
    {
        var item = new ThreeResultMatrix {FirstMatrix = value[0], SecondMatrix = value[1]};
        var startTime :long = DateTime.Now.ToFileTimeUtc();
        _matrixCalculationService.MatrixMultiplication(item);
        item.Time = DateTime.Now.ToFileTimeUtc() - startTime;
        return new JsonResult(item);
    }
    catch (Exception ex)
    {
        return BadRequest(ex);
    }
}
```

Результат:

Matrix A					Matrix B				
3	1	2	1	3	3	3	2	1	1
1	3	3	1	1	2	2	1	2	2
2	2	2	2	1	2	3	2	2	1
3	1	3	2	2	2	1	3	2	2
1	1	2	2	2	3	2	2	3	2
A * B									
26	24	20	20	15					
20	21	16	18	14					
21	20	18	17	14					
27	26	23	21	16					
19	17	17	17	13					
317877									
Row					<input type="text" value="5"/>				
Column					<input type="text" value="5"/>				
<input type="button" value="Генерація матриць"/>									
<input type="button" value="Відправити значення"/>					<input type="button" value="Без значень"/>				

5. Різниця в часі в даному випадку буде залежати тільки від розмірності матриць. Чим більші матриці ми відправляємо на розрахунок серверу тим довше буде йти запит на сервер. Якщо ж сервер бере дані з власних ресурсів цим часом можна знехтувати.

Висновок: В ході даної лабораторної роботи був розроблений клієнт-серверний додаток, для підрахунку матриць.

Також було на прикладі коду виявлено різницю в часі який буде потрібний для виконання операцій.