

Лабораторная работа № 2

Технологии распределенной обработки данных

Тема: Разработка распределённой системы обработки данных

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

Проверил:

Гай В. Е.

Выполнил:

Студент гр. 14-В-2

Самсонов И. А.

## 1. Цель и порядок выполнения работы

Разработка распределённой системы обработки данных.

Разработанный программный комплекс должен состоять из Сервера и Клиента.

Функции сервера: хранение удалённого объекта, предоставляющего доступ к заданиям для обработки и результату обработки.

Предусмотреть на сервере возможность одновременного доступа к критической секции кода нескольких клиентов (с помощью оператора lock). Критическая секция кода - та, к которой гипотетически одновременно могут обратиться несколько клиентов.

Функции клиента (на сервере хранится список клиентов - эта функция уже предусмотрена исходным кодом библиотеки RemoteBase):

1) управляющие функции (выполняет только один клиент из всего множества клиентов, выполнение данной функции должно выполняться через вызов методов удалённого объекта (удалённый объект хранится на сервере)):

1.1) формирование и ведение списка заданий (под ведением понимается удаление уже обработанных и предоставление клиенту задания по запросу);

1.2) получение, объединение и вывод результатов вычислений (результаты вычислений должны выводиться в каждом клиенте, для этого необходимо проверять окончание обработки всех данных по таймеру; объединение

результатов вычисление также можно реализовать с использованием таймера);

1.3) устанавливает флаг того, что управляющий клиент назначен, на сервере сохраняется идентификатор клиента;

2) вычислительные функции

2.1) запрос задания с сервера (клиент должен запросить задание только после того, как эти задания были сформированы);

2.2) обработка данных;

2.3) отправка результатов обработки на сервер.

Требования к системе:

1) предусмотреть возможность отключения одного из клиентов, получившего задание на обработку.

2) предусмотреть возможность отключения управляющего клиента (для этого можно хранить время последней операции на сервере).

## Лабораторная работа № 2

Разработка распределённой системы обработки данных

14-B-2

Содержание

- 1. Листинг RemoteClient
- 2. Листинг RemoteBase
- 3. Листинг RemoteServer

Инв. № подл	Подп. и дата				Инв. № дубл.	Взам. инв. №	Подп. и дата
Ли	Изм.	№ докум.	Подп.	Дат			
					Лист		

## 2. Выполнение лабораторной работы

### 2.1. Листинг RemoteClient

```
using System;
using System.Windows.Forms;

using NetRemotingLibrary;

using System.Runtime.Remoting.Channels.Tcp;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Services;
using System.Runtime.Remoting;

namespace NetRemotingClient
{
    class Program
    {
        const int SYS_PORT = 9232;
        const string SYS_LIB_NAME = "Lib";

        private static RemotingLibrary remotingClass = null;

        static void Main(string[] args)
        {
            TcpChannel clientChannel = new TcpChannel();
            ChannelServices.RegisterChannel(clientChannel, true);

            string connectingString = string.Format("tcp://localhost:{0}/{1}", SYS_PORT, SYS_LIB_NAME);
            remotingClass = (RemotingLibrary)Activator.GetObject(typeof(RemotingLibrary), connectingString);

            if (remotingClass == null)
            {
                Console.WriteLine("Client not connected! Abort!");
                return;
            }

            bool isAdmin = false;
            int clientID = remotingClass.RegisterClient(out isAdmin);

            // admin
            if (isAdmin)
            {
                Console.WriteLine("Current client connected as admin.");

                const int rows = 200;
                const int cols = 200;

                int[,] task = new int[rows, cols];
                Random r = new Random();
                for (int i = 0; i < rows; ++i)
```

Инв. № подл	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	
Ли	Изм.	№ докум.	Подп.	Дат	Лабораторная работа № 2
					Лист 3

```

for (int j = 0; j < cols; ++j)
task[i, j] = r.Next(10);

Console.WriteLine("Uploading task...");
remotingClass.UploadTaskToServer(clientID, task);

Console.WriteLine("Task uploaded.");

Console.WriteLine("Press any key to disconnect admin client...");
Console.ReadKey();

Console.WriteLine("Disconnecting...");
remotingClass.UnregisterClient(clientID);

Console.WriteLine("Ended.");
return;
}

Console.WriteLine("Current client connected as processor client.");

while (!remotingClass.isWorkFinished())
{
Console.WriteLine();

int[] array = remotingClass.GetClientData(clientID);

Console.WriteLine("Client received data: ");
for (int i = 0; i < array.GetLength(0); ++i)
Console.Write("{0} ", array[i]);
Console.WriteLine();

Console.WriteLine("Sorting...");

for (int i = 0; i < array.GetLength(0); ++i)
{
for (int j = i; j < array.GetLength(0); ++j)
{
if (array[i] > array[j])
{
int a = array[i];

array[i] = array[j];
array[j] = a;
}
}
}

Console.WriteLine("Client sorted data: ");
for (int i = 0; i < array.GetLength(0); ++i)
Console.Write("{0} ", array[i]);
Console.WriteLine();

remotingClass.ReturnClientData(clientID, array);
}

remotingClass.UnregisterClient(clientID);

Console.WriteLine("Client finished.");
}
}
}

```

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Инв. № подл

Лист

4

Лабораторная работа № 2

Ли Изм. № докум. Подп. Дат

## 2.2. Листинг RemoteBase

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace NetRemotingLibrary
{
    public class Client
    {
        public enum ClientStatus { BUSY, FREE }

        private int id = 0;
        private ClientStatus status = ClientStatus.FREE;
        private bool hadAdminRights = false;
        private DateTime lastInteraction = DateTime.Now;
        private int meta = 0;

        public Client(int clientID)
        {
            id = clientID;
            status = ClientStatus.FREE;
            hadAdminRights = false;
        }

        public void SetAdmin(bool admin)
        {
            hadAdminRights = admin;
        }

        public bool GetAdmin()
        {
            return hadAdminRights;
        }

        public int GetID()
        {
            return id;
        }

        public void SetStatus(ClientStatus clientStatus)
        {
            status = clientStatus;
            lastInteraction = DateTime.Now;
        }

        public DateTime GetTimeSinceLastInteraction()
        {
            return lastInteraction;
        }

        public ClientStatus GetStatus()
        {
            return status;
        }

        public int GetMeta()
        {
            return meta;
        }
    }
}
```

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Инв. № подл

Лист

5

Лабораторная работа № 2

Ли Изм. № докум. Подп. Дат

```

public void SetMeta(int clientMeta)
{
    meta = clientMeta;
}
}

public class RemotingLibrary : MarshalByRefObject
{
    public static int[] Sort(int[] array)
    {
        return array;
    }

    private static int globalID = 0;
    private static List<Client> clients = new List<Client>();

    private int[,] serverTask;
    private bool[] rowProcessed;
    private int[] clientData;

    private System.Timers.Timer timer = new System.Timers.Timer();

    public RemotingLibrary()
    {
        timer.Elapsed += timer_Elapsed;
        timer.Interval = 1000;
        timer.Start();
    }

    void timer_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
    {
        lock ("timer")
        {
            for (int i = clients.Count - 1; i >= 0; —i)
            {
                if ((clients[i].GetStatus() == Client.ClientStatus.BUSY) && (DateTime.Now - clients[i].GetTimeSinceLastAccess() > 10))
                {
                    Console.WriteLine("Client {0} disconnects!", clients[i].GetID());

                    int rowID = clients[i].GetMeta();
                    rowProcessed[rowID] = false;

                    clients.RemoveAt(i);
                }
            }
        }

        private Client GetClientByID(int clientID)
        {
            Client client = null;
            for (int i = 0; i < clients.Count; ++i)
            {
                if (clients[i].GetID() == clientID)
                {
                    client = clients[i];
                }
            }

            return client;
        }

        public int RegisterClient(out bool isAdmin)

```

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Инв. № подл

Лист

6

Лабораторная работа № 2

Ли Изм. № докум. Подп. Дат

```

{
lock ("client_connect_request")
{
globalID++;

Client client = new Client(globalID);

bool hasAdmin = false;
for (int i = 0; i < clients.Count; ++i)
hasAdmin |= clients[i].GetAdmin();

if (!hasAdmin)
{
client.SetAdmin(true);
isAdmin = true;
}
else
isAdmin = false;

clients.Add(client);

return globalID;
}
}

public void UnregisterClient(int clientID)
{
lock ("unregister_client")
{
for (int i = 0; i < clients.Count; ++i)
if (clients[i].GetID() == clientID)
clients.RemoveAt(i);
}
}

public void UploadTaskToServer(int clientID, int[, ] task)
{
lock ("admin_upload_task")
{
Client client = GetClientByID(clientID);

if ((client == null) || (!client.GetAdmin()))
{
Console.WriteLine("[ERROR] Only admin can give task's to server.");
return;
}

if ((rowProcessed != null) && (!isWorkFinished()))
{
Console.WriteLine("Server not fully completed previous work. Aborting.");
return;
}

Console.WriteLine();
Console.WriteLine("Admin [client ID: {0}] upload task to server...", clientID);

serverTask = new int[task.GetLength(0), task.GetLength(1)];
rowProcessed = new bool[task.GetLength(0)];
clientData = new int[task.GetLength(1)];

Console.WriteLine("Copying task...");

```

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл	

Ли	Изм.	№ докум.	Подп.	Дат



```

for (int i = 0; i < task.GetLength(0); ++i)
{
    for (int j = 0; j < task.GetLength(1); ++j)
        serverTask[i, j] = task[i, j];

    rowProcessed[i] = false;
}

Console.WriteLine("Task copied");
}
}

public int[] GetClientData(int clientID)
{
    lock ("data_request")
    {
        Client client = GetClientByID(clientID);
        int i = 0;
        for (; i < rowProcessed.Length; ++i)
            if (rowProcessed[i] == false)
                break;

        for (int j = 0; j < serverTask.GetLength(1); ++j)
            clientData[j] = serverTask[i, j];

        rowProcessed[i] = true;
        client.SetStatus(Client.ClientStatus.BUSY);
        client.SetMeta(i);

        return clientData;
    }
}

public void ReturnClientData(int clientID, int[] data)
{
    lock ("data_return")
    {
        Client client = GetClientByID(clientID);
        int i = client.GetMeta();

        for (int j = 0; j < serverTask.GetLength(1); ++j)
            serverTask[i, j] = data[j];

        client.SetStatus(Client.ClientStatus.FREE);

        bool flag = true;
        for (i = 0; i < clients.Count; ++i)
            flag &= (clients[i].GetStatus() == Client.ClientStatus.FREE);

        // Server ended
        if (flag && isWorkFinished())
        {
            Console.WriteLine("Server ended task");
            Console.WriteLine("Result is: ");
            for (i = 0; i < serverTask.GetLength(0); ++i)
            {
                for (int j = 0; j < serverTask.GetLength(1); ++j)
                    Console.Write("{0} ", serverTask[i, j]);

                Console.WriteLine();
            }
        }
    }
}

```

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

Инв. № подл

Лабораторная работа № 2

Лист

8

Ли Изм. № докум. Подп. Дат

```

}
}
}

public bool isWorkFinished()
{
    lock ("check_work_finished")
    {
        bool flag = true;
        for (int i = 0; i < rowProcessed.Length; ++i)
            flag &= rowProcessed[i];

        return flag;
    }
}
}
}
}

```

### 2.3. Листинг RemoteServer

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Runtime.Remoting.Channels.Tcp;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Services;
using System.Runtime.Remoting;

namespace NetRemotingServer
{
    class Program
    {
        const int SYS_PORT = 9232;
        const string SYS_LIB_NAME = "Lib";

        static void Main(string[] args)
        {
            Console.WriteLine("Server start ...");

            Console.WriteLine("Opening {0} port for listening...", SYS_PORT);

            TcpChannel channel = new TcpChannel(SYS_PORT);

            Console.WriteLine("Registering channel with security ensuring");
            ChannelServices.RegisterChannel(channel, true);

            Console.WriteLine("Pass library to memory...");
            RemotingConfiguration.RegisterWellKnownServiceType(
                typeof(NetRemotingLibrary.RemotingLibrary), SYS_LIB_NAME, WellKnownObjectMode.Singleton);

            Console.WriteLine("Press <ENTER> to shutdown server.");
            Console.ReadLine();

        }
    }
}

```

### 2.4. Тестирование работоспособности системы

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<div>Лабораторная работа № 2</div>	Лист 9
	Ли	Изм.	№ докум.	Подп.	Дат	

```
C:\WINDOWS\system32\cmd.exe

Client received data:
0 8 1 2 7 8 5 7 8 5 7 8 3 2 6 1 8 8 5 0
Sorting...
Client sorted data:
0 0 1 1 2 2 3 5 5 5 6 7 7 7 8 8 8 8 8 8

Client received data:
4 0 1 0 8 1 0 8 9 4 2 3 9 7 9 5 5 0 2 8
Sorting...
Client sorted data:
0 0 0 0 1 1 2 2 3 4 4 5 5 7 8 8 8 9 9 9

Client received data:
7 4 7 9 6 7 7 9 6 2 2 2 4 7 9 0 6 0 4 6
Sorting...
Client sorted data:
0 0 2 2 2 4 4 4 6 6 6 6 7 7 7 7 7 9 9 9

Client received data:
8 6 8 4 3 3 8 6 3 0 6 7 6 7 3 9 2 1 3 0
Sorting...
Client sorted data:
0 0 1 2 3 3 3 3 3 4 6 6 6 6 7 7 8 8 8 9
```

Рисунок 2.1. Пример работы клиента

```
Server

Copying task...
Task copied
Server ended task
Result is:
0 0 2 2 3 3 3 4 5 5 6 6 6 6 7 8 8 9 9 9
0 0 0 1 1 1 2 2 2 3 4 4 5 5 6 7 7 7 7 9
0 0 0 1 1 2 3 4 4 5 7 7 7 7 8 8 9 9 9
0 0 0 1 2 3 3 4 4 5 5 6 6 6 8 8 8 8 9
0 0 1 1 2 2 2 2 2 2 3 4 5 5 6 6 6 6 6 8
0 0 0 1 1 1 2 2 3 4 5 5 7 7 7 7 8 8 9 9
0 0 1 2 2 2 2 2 3 4 5 5 5 6 7 7 8 9 9 9
0 1 1 1 2 3 3 4 4 4 5 6 6 7 7 9 9 9 9 9
0 1 1 1 2 2 2 3 3 4 4 4 5 6 6 6 7 8 8
0 0 0 1 2 2 2 2 3 3 4 4 5 6 6 6 6 7 9
0 0 1 1 2 2 3 5 5 5 6 7 7 7 8 8 8 8 8
0 0 0 0 1 1 2 2 3 4 4 5 5 7 8 8 8 9 9 9
0 0 2 2 2 4 4 4 6 6 6 6 7 7 7 7 9 9 9
0 0 1 2 3 3 3 3 3 4 6 6 6 6 7 8 8 8 9
0 0 1 2 2 3 3 4 4 5 6 6 6 6 7 8 8 9 9
0 0 1 2 2 3 3 4 4 5 5 5 5 6 6 6 6 7 8 8
0 0 0 1 1 1 2 2 3 3 3 4 4 5 6 6 6 7 8 8
0 0 1 1 2 2 2 2 3 3 3 4 4 5 5 7 8 8 8
1 1 1 2 2 3 4 4 5 6 6 6 7 8 9 9 9 9 9
0 0 1 1 1 2 2 3 3 5 5 6 6 7 8 8 9 9 9
```

Рисунок 2.2. Пример работы сервера

## 2.5. Вывод

В результате выполнения данной лабораторной работы был изучен компонент созданный фирмой Microsoft для межпроцессного взаимодействия .NET Remoting. На основе этого компонента была реализована простейшая сиситема распределенной обработки данных реализующая сортировку методом пузырька.

Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	
Инв. № подл	

Ли	Изм.	№ докум.	Подп.	Дат