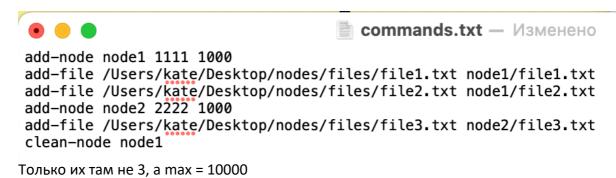
Отчет лаба 4, Жаркова Е.С.

## Суть лабы:

У нас есть файлик с командами, которые должны, как ни странно, выполняться:



Запускаем клиенты на трех нодах с разными портами и запускаем сервак, который пробежится по командам и выполнит их.

В program.cs/server мы запускаем Exec(принимает путь), который в свою очередь выполняет команды, описанные в commands.txt Команды делятся на два типа (по моему скромному мнению):

С которыми все просто, и не нужно писать алгоритм: Add-node Add-file Remove-file

У которых нужно половить два дня ошибки: Clean-node Balance-node

А теперь посмотрим, насколько плохо это работает:

```
Call Tree
      ▼ 100 % All Calls • 349 MB

y 99.6 

Main • 348 MB • Server. Program. Main()

Main • 348 MB • Server. Program. Main()

✓ 348 MB • Server. Program. Main()

Main • 348 MB • Server. Program.

Main • 348 MB • Server.

Main • 348 MB • Serve
                       99.6 % Exec • 348 MB • Server. Commands. Exec (String)

y 99.0 % GoCommand • 346 MB • Server. Commands. GoCommand (String)

✓ 59.4 % AddFile • 208 MB • Server. Commands. AddFile (String, String)

                                         > 58.6 % AddFile • 205 MB • Server. TransferByte. AddFile (String, String)
                                                 0.06 % Substring • 0.2 MB • System.String.Substring(Int32, Int32)
                                26.0 % BalanceNode • 91 MB • Server.Commands.BalanceNode()
                                        > 25.0 % AddFile • 87 MB • Server. Commands. AddFile (String, String)
                                        > 0.52 % RemoveFile • 1.8 MB • Server. Commands. RemoveFile (String)
                                                          CleanNode • 42 MB • Server. Commands. CleanNode (String)
                                         > 10.8 % AddFile • 38 MB • Server.Commands.AddFile(String, String)
                                        > 0.78 % RemoveFile • 2.7 MB • Server. Commands. RemoveFile (String)
                                                 0.44 % Concat • 1.5 MB • System. String. Concat(String, String, String)
                                          > 0.08 % AddFile • 0.3 MB • Server. Node. AddFile (File)
```

```
Call Tree

✓ ▼ 100 % All Calls • 18,825 ms

✓ 49.5 % Main • 9,323 ms • 1 call • Server.Program.Main()

✓ 49.5 % Exec • 9,321 ms • 1 call • Server.Commands.Exec(String)

✓ 49.2 % GoCommand • 9,262 ms • 10,005 calls • Server.Commands.GoCommand(String)

> 30.9 % AddFile • 5,815 ms • 10,000 calls • Server.Commands.AddFile(String, String)

> 9.58 % BalanceNode • 1,803 ms • 1 call • Server.Commands.BalanceNode()

> 7.69 % CleanNode • 1,448 ms • 1 call • Server.Commands.CleanNode(String)

> 0.39 % AddNode • 73 ms • 3 calls • Server.Commands.AddNode(String, Int32, Int32)

▼ 0.31 % WriteLine • 58 ms • 10,000 calls • System.Console.WriteLine(String)

▼ 0.06 % TrimWhiteSpaceHelper • 11 ms • 48,850 calls • System.String.TrimWhiteSpaceHelper(TrimType)

▼ 0.04 % Parse • 7.1 ms • 6 calls • System.Int32.Parse(String)

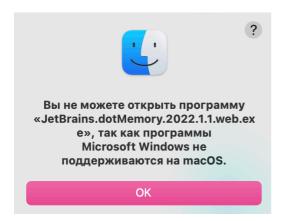
▼ 0.04 % Remove • 6.6 ms • 20,011 calls • System.String.Remove(Int32, Int32)

▼ 0.03 % Substring • 6.2 ms • 20,011 calls • System.String.Substring(Int32, Int32)
```

Посмотрели, теперь думаем, как это можно улучшить...

0) Для начала, как все нормальные люди, удаляем Скайп.

Теперь прекращаем шутить и открываем dotTrace Еще пытаемся скачать dotMemory, вспоминаем, что работаем на маке и плачем в сторонке



1) Заменили class'ы на record'ы, но это дало мало...

```
Call Tree

▼ 100 % All Calls • 349 MB
       99.6 % Main • 348 MB • Server. Program. Main()
               99.6 % Exec • 348 MB • Server. Commands. Exec (String)
                      98.9 % GoCommand • 346 MB • Server.Commands.GoCommand(String)

y 59.4 % AddFile • 207 MB • Server. Commands. AddFile (String, String)

√ 39.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

√ 39.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

√ 39.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

√ 39.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

√ 39.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

✓ 49.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

✓ 49.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

✓ 49.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

✓ 49.4 % AddFile • 207 MB • Server. Commands. AddFile (String)

✓ 49.4 % AddFile • 209 MB • Server. Commands.

✓ 49.4 % AddFile • 209 MB • 209 MB • 200 
                                      > 58.3 % AddFile • 204 MB • Server. TransferByte. AddFile (String, String)
                                              ▼ 0.15 % FileInfo..ctor • 0.5 MB • System.IO.FileInfo..ctor(String)
                                              ▼ 0.15 % GetFileName • 0.5 MB • System.IO.Path.GetFileName(String)
                                      > 0.09 % AddFile • 0.3 MB • Server. Node. AddFile (File)
                                              0.09 % Substring • 0.3 MB • System. String. Substring (Int32, Int32)
                              > 25.9 % BalanceNode • 91 MB • Server.Commands.BalanceNode()
                              > 12.1 % CleanNode • 42 MB • Server. Commands. CleanNode (String)
                                      ▼ 0.41 % TrimWhiteSpaceHelper • 1.4 MB • System.String.TrimWhiteSpaceHelper(TrimType)
                                      ▼ 0.34 % Substring • 1.2 MB • System.String.Substring(Int32, Int32)
                                     0.23 % Concat • 0.8 MB • System. String. Concat (String, String)
                                      ▼ 0.06 % Trim • 0.2 MB • System. String. Trim()
                              > 0.03 % AddNode • 0.1 MB • Server. Commands. AddNode (String, Int32, Int32)
```

2) Меняем на arrayPool, это значительноооооо помогло

```
Call Tree

∨ ▼ 100 % All Calls • 41 MB

∨ 96.7 % Main • 40 MB • Server.Program.Main()

∨ 96.7 % Exec • 40 MB • Server.Commands.Exec(String)

∨ 88.0 % GoCommand • 36 MB • Server.Commands.GoCommand(String)

∨ 46.2 % AddFile • 19 MB • Server.Commands.AddFile(String, String)

> 37.5 % AddFile • 15 MB • Server.TransferByte.AddFile(String, String)

▼ 1.98 % FileInfo..ctor • 0.8 MB • System.IO.FileInfo..ctor(String)

▼ 1.23 % GetFileName • 0.5 MB • System.IO.Path.GetFileName(String)

> 0.27 % AddFile • 0.1 MB • Server.Node.AddFile(File)

0.25 % Substring • 0.1 MB • System.String.Substring(Int32, Int32)

> 27.2 % CleanNode • 11 MB • Server.Commands.CleanNode(String)

▼ 6.19 % TrimWhiteSpaceHelper • 2.5 MB • System.String.TrimWhiteSpaceHelper(TrimType)

3.95 % Remove • 1.6 MB • System.String.Remove(Int32, Int32)

2.22 % Concat • 0.9 MB • System.String.Remove(Int32, Int32)

> 0.24 % AddNode • 0.1 MB • Server.Commands.AddNode(String, Int32, Int32)

> 0.24 % AddNode • 0.1 MB • Server.Commands.AddNode(String, Int32, Int32)
```

```
Call Tree

✓ ▼ 100 % All Calls · 12,768 ms

✓ 49.2 % Main · 6,286 ms · 1 call · Server.Program.Main()

✓ 49.2 % Exec · 6,284 ms · 1 call · Server.Commands.Exec(String)

✓ 48.9 % GoCommand · 6,238 ms · 10,005 calls · Server.Commands.GoCommand(String)

> 25.9 % AddFile · 3,310 ms · 10,000 calls · Server.Commands.AddFile(String, String)

> 11.3 % CleanNode · 1,438 ms · 1 call · Server.Commands.CleanNode(String)

> 10.1 % BalanceNode · 1,285 ms · 1 call · Server.Commands.BalanceNode()

> 0.57 % AddNode · 72 ms · 3 calls · Server.Commands.AddNode(String, Int32, Int32)

▼ 0.46 % WriteLine · 59 ms · 10,000 calls · System.Console.WriteLine(String)

▼ 0.13 % TrimWhiteSpaceHelper · 16 ms · 48,860 calls · System.String.TrimWhiteSpaceHelper(TrimType)

▼ 0.07 % Remove · 9.0 ms · 20,011 calls · System.String.Remove(Int32, Int32)
```

3) Заменяем на сложные конструкции \$"{переменная}/{переменная}" на stringBuilder Улучшили!

```
Call Tree

✓ ▼ 100 % All Calls • 30 MB

✓ 95.5 % Main • 28 MB • Server.Program.Main()

✓ 95.5 % Exec • 28 MB • Server.Commands.Exec(String)

✓ 86.2 % GoCommand • 26 MB • Server.Commands.GoCommand(String)

✓ 65.2 % AddFile • 19 MB • Server.Commands.AddFile(String, String)

> 56.4 % AddFile • 17 MB • Server.TransferByte.AddFile(String, String)

▼ 1.70 % FileInfo..ctor • 0.5 MB • System.IO.FileInfo..ctor(String)

1.02 % Substring • 0.3 MB • System.String.Substring(Int32, Int32)

▼ 0.34 % GetFileName • 0.1 MB • System.IO.Path.GetFileName(String)

6.11 % Substring • 1.8 MB • System.String.Substring(Int32, Int32)

▼ 5.43 % TrimWhiteSpaceHelper • 1.6 MB • System.String.TrimWhiteSpaceHelper(TrimType)

5.43 % Remove • 1.6 MB • System.String.Remove(Int32, Int32)

3.73 % Concat • 1.1 MB • System.String.Concat(String, String)

> 0.33 % AddNode • 0.1 MB • Server.Commands.AddNode(String, Int32, Int32)

▼ 9.23 % ReadLine • 2.8 MB • System.IO.StreamReader.ReadLine()
```

4) Самое интересное! Переделываем парсинг команд с

```
public void GoCommand(string str)
    string command = str.Substring(startIndex: 0, length: str.IndexOf(' ')).Trim();
    str = str.Remove(startIndex: 0, count: str.IndexOf(' ')).Trim();
    string nodeName, path, partialPath;
    switch (command)
        case "add-node":
            nodeName = str.Substring(startIndex: 0, length: str.IndexOf(' ')).Trim();
            string portStr = str.Substring(startIndex: 0, length: str.IndexOf(' ')).Trim();
            str = str.Remove(startIndex: 0, count: str.IndexOf(' ')).Trim();
            int port = Int32.Parse(portStr);
            string sizeStr = str.Trim();
            int size = int.Parse(sizeStr);
            AddNode(nodeName, port, size);
            break;
        case "add-file":
            path = str.Substring(startIndex: 0, length: str.IndexOf(' ')).Trim();
            str = str.Remove(startIndex: 0, count: str.IndexOf(' ')).Trim();
            partialPath = str.Trim();
            Console.WriteLine($"add file {partialPath}");
            AddFile(path, partP: partialPath);
            break;
```

```
var commands:string[] = str.Split();
string command = commands[0];
string nodeName, path;
switch (command)
    case "add-node":
        nodeName = commands[1];
       string portStr = commands[2];
        int port = Int32.Parse(portStr);
        string sizeStr = commands[3];
        int size = int.Parse(sizeStr);
        AddNode(nodeName, port, size);
        break;
    case "add-file":
        path = commands[1];
        var partialPath:string = commands[2];
        AddFile(path, partP: partialPath);
        break;
```

## И получаем:

```
Call Tree

✓ ▼ 100 % All Calls • 58 MB

✓ 97.7 % Main • 57 MB • Server.Program.Main()

✓ 97.7 % Exec • 57 MB • Server.Commands.Exec(String)

✓ 91.8 % GoCommand • 53 MB • Server.Commands.GoCommand(String)

→ 32.5 % AddFile • 19 MB • Server.Commands.AddFile(String, String)

→ 32.4 % CleanNode • 19 MB • Server.Commands.CleanNode(String)

→ 22.9 % BalanceNode • 13 MB • Server.Commands.BalanceNode()

▼ 3.84 % SplitInternal • 2.2 MB • System.String.SplitInternal(ReadOnlySpan, Int32, StringSplitOptions)

→ 0.17 % AddNode • 0.1 MB • Server.Commands.AddNode(String, Int32, Int32)

▼ 5.91 % ReadLine • 3.4 MB • System.IO.StreamReader.ReadLine()
```