

PLP Final Assignment Report

Eyad Mohammed Osama Bereh - 201620054

The idea of the assignment is to read a file and distribute its contents across a specified number of processes using MPI.NET.

Here's the output of command **hostname**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>hostname
DESKTOP-PL99CBD

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>
```

Here's the output of command **date**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>date
The current date is: Sun 07/19/2020
Enter the new date: (mm-dd-yy)

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>
```

Here's the output of command **time**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>time
The current time is: 20:04:17.85
Enter the new time:

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>
```

Here's the output of command **mpiexec**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec
Microsoft MPI Startup Program [Version 10.1.12498.18]

Launches an application on multiple hosts.

Usage:

    mpiexec [options] executable [args] [ : [options] exe [args] : ... ]
    mpiexec -configfile <file name>

Common options:

-n <num_processes>
-env <env_var_name> <env_var_value>
-wdir <working_directory>
-hosts n host1 [m1] host2 [m2] ... hostn [mn]
-cores <num_cores_per_host>
-lines
-debug [0-3]
-logfile <log file>

Examples:

    mpiexec -n 4 pi.exe
    mpiexec -hosts 1 server1 master : -n 8 worker

For a complete list of options, run mpiexec -help2
For a list of environment variables, run mpiexec -help3

You can reach the Microsoft MPI team via email at askmpi@microsoft.com
```

As for the input text file, i didn't use any text file.

Instead, i've used **PHP** programming language and wrote a script that generates random lines of characters and numbers and stored it in a file, then i'm loading this file into the parallel program.

In fact, here's the content of this script:

```
1 <?php
2 $number_of_lines = $argv[1];
3 $line_length_min = $argv[2];
4 $line_length_max = $argv[3];
5
6 $characters = [ "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O",
7                "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "a", "b", "c", "d",
8                "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s",
```

```
9      "t", "u", "v", "w", "x", "y", "z", 0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
10
11      $str = "";
12      echo count($characters);
13      for ($i = 0; $i < $number_of_lines; $i++) {
14          $line_length = rand($line_length_min, $line_length_max);
15          for ($j = 0; $j < $line_length; $j++) {
16              $index = rand() % count($characters);
17              $str .= $characters[$index];
18          }
19          $str .= "\n";
20      }
21
22      echo $str;
23      ?>
```

For example, if we executed the command **php text-generator.php 30 70 100 > input.txt**, we will get a text file with **lines** of random characters and numbers of length between **70** and **100**, something like the following:

```
62FgeYVye4CBqBP3L19k5nP222de8xZEgin1YuLDCyx09E8T9rx9nbaPpjdyM4ZQ44wdkEGSmmGQpD7e56yDN53AtsCNJG
NIzd1xJ3aaRjJYsP6u3X0bsXc7BDdzVQRwh0vGJTT11BU7WC2G0tgLgYiLWdTrNGJtFZTzJiCWcdM
QTuYh3oLZXiN3B1GuA3Ig8RK38B2hcISxT3mUYK6glavoA8vXwjWf7JzcxHtOG67kcNwyeTsr
g9CFrZf13JmvRlMXNtLZFY21uinEuX0p57qqzmqqbbGdVkZ2nQcZSNTv0dCDlTwoVfJ7N5bVlZV6Qwo
Go5RCsVb5xS1CEnG1bEgj509QsG5a5irvDRfj5x5D5EfoWlm9D5wkFmChc0GEP6Dz6J8HmN2lR7E1gue6hHUdbnaU6m
7S8CEcYLXB5Uc33cZL50Z6okvMWs8kZIGxaeTLfEJo30jfxqKn6K0n6qC4XTyFW7YX9CwdnWmmqH6Cg8hWw4i
SsH0ND0YvE8dIYg6pD4q5v5MZdd0xNwTAH9DDvzL29lUUnIBuWR1mX93hzkIlplw1plt7W7Wj1aVkm5ljZmW4P4CG
ou2f6dCTV9lvG5bWRuGZjN308cabAjQL7IyuE50rG148at2SwB2T0sKF0GkYTRp8JaiDeMEZfCm8tzrYTqM
FoMkqh3WWQ1bajqP3Jn1NexVmWKzr557XtK6L6acyHbrdjBKKSwmLV1x3XdDdraJU2CSXEH
PrzF3UAW9qt6iwEN4C2zfxrVDeqpPdk7tADDXKv2UjW83Qiihes62nWcnKFg3gS3jl5FZDYHT0vJYPNqKrZm2ssqM3YeVcr3QPAR
6ygSHCTR75ZqpWA3EISzZid5p7zgbVnscZhs20nk7j2Yn14u2SLdNKRutDHESbBpvXdMJqLqsPxXsRJnHJ0il15j9L2wW2aarATp
N7PV5sakD7hFThpW1acX6eDLfuweXW7qYezjFuH0dvBVVQV283J10hwaJyGkcEwBs08jFYQRagE
Byy3ZPikDNeX1I219Y5FBmZYYa4cjbAnpa5fGpHLzeGKypNWpSpkEI1HuCnTYotTjsCeNVZU3bZ3a1X7Jm419Q
kZSEfjulKocTbQZkWejHvprXNA2zC6Befj5pZHHAmgP0XRZQb4u5o2q1n9tVuRW2pMSBgyDpYPX
YWMln1CEgen3tG0HwaFT1lhZFHAIA0A0njw0YrqcN636eTW35inBURC955I0sLMoVnKaKof5xhJ
sS6kcVD2t3xAf6PEmaYRABj9pxkKXxs3q2Tb7teGM8eGHu1ftXTV1vbzPmlGbryr4i4fdIulj6EV9xZ
aVZQuZSaeMB56L1kgEPdEgI4sm6cxorEcyNa6Xoy8jpkZXKuTavxT0xmaz1wclPns4BtfAB0rP
jgpplcuH2A5V37ya0AnMyz1ldp9x2pL6HV3JvX7zW4Utf2fNkk1eFEn60WGKNuzxENSqG8Fjs7sMRPBjXNW5z
kNDiFQtBA7TfRRK3PQM100DFSXJs2WImUj5eYYL5a5drKp20v0a3B2DW7ZYWoVFYHw0GDBAWVCSgy
gqD0ru747a9GLQ4DocKEA5wJjs00gt0x6igtrVozE0UDWcceSYI90dStNPZj0uDqu63YbZaEsPIkF5q0BWdTxE
Ws73LDhpsbAVjmq1RARt9z7mUYkZM2JmQTr7IC7bp0rP2xeyP3natF8birEn6BBM0qun84Qa6kemunfMT7IGAhsq8
JcMr10lraQysdGQH3m330bHD2pBLl0nWrS0ryGy9js7odBiviU4y9rPC5ybUEyWQGGFP4eobDzXInHTfliKTcqucUHKTXCDH
f1bf90vEz8okIr7T0nKQ5a1R0Y6UA8Ln4h71ffr1Kqojm17LyJsJ6ibesBZzcMxpsH1VTTa7LBqBjUb8eNWVleyWX96DLmvEzTO
T0brVoW4VQycfudzkkFAE0QXiTJvJbbRkgEqCwjvUd8x74wa8DKl8LFxnapUIiML4uTvTbmVjHmX
JYKtdgguhe0Ws99dm7RJnD0qkMt0QANQ2uTWXYFFdI0p0jaNBA97F6GpQWP9Rb3S0ZRWqlW91kiK593eQL9koW2PEugu
lfipN0g1wEAgAsJ87X0ZofiQN2kEKzpY7LsagtHr3GHmCTpksARSSZ3GRjZ003VD4nrK7Y09MetRvDZNJ0sI1W05iq
2hvpvXKLAKwCMGex0jKVehuxTSNDurAXKQ9BJJbnvkQVKJEF2cXq2ABS8CcSszA8bu9KXrCwd31tkDSjlMjfZxicZq
5nova0eWJvToowD1XWSJL3PPeTWkw4mboxocS0aYl2n4pFr0sVGV0Q8wU0GqegeyG9P14VGHb6k7r8GKVhcn2VQofPN8qSu9WFG
CmPL50qTJ0QtCn1PD70vnJJBbYrXYmU43egCwoCEivqqTbYlY1k0IxbAPRASID6oUNEgR7pJi1GCnXwSmw3f
BlBJVEWqaxRVRw1ls3M05jKPithAlje5lsh4exWUnMnKkCCsDnYwkdC2e5Di4VFHAJyD8Kr1p0GHXJLL
```

And here's the content of **C#** program alongside its comments:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Diagnostics;
4 using System.IO;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8 using MPI;
9 using MPIUtils;
10
11 namespace PLP_Assignment
12 {
13     class Program
14     {
15         private const string PATH = @"input.txt"; // Path to text file
16         private const int ROOT_PROCESS_ID = 0; // Define Root Process ID
17         public static int Main(string[] args)
18         {
19             using (new MPI.Environment(ref args)) // Initialize MPI Environment
20             {
21                 Intracommunicator comm = Communicator.world; // Define a shortcut to world communicator
22                 string[] contents = null; // Right here, we will store the lines of the text file
23                 char what = '\0'; // The character which the user enters
24
25                 int totalCharacters = 0; // Total characters per process
26                 int totalOccurrences = 0; // Total occurrences per process
27                 double totalTime = 0; // Total time per process
28                 bool verbose = false; // Verbose output on command line
29                 if (args.Length > 0 && args[0].ToLower() == "--verbose") // If the flag "--verbose" exists, set verbose = true
30                 {
31                     verbose = true;
32                 }
33
34                 int allCharacters; // All characters by all processes
35                 int allOccurrences; // All occurrences by all processes
36
37                 // If we're in the root process
38                 if (ProcessID == 0)
39                 {
40                     // Read the contents of the text file
41                     contents = File.ReadAllLines(PATH);
42                     // Get the length of the text file
43                     totalCharacters = contents.Sum(line => line.Length);
44                     // Get the total occurrences of each character
45                     totalOccurrences = contents.Sum(line => line.Count(c => c == what));
46                     // Get the total time
47                     totalTime = Stopwatch.GetElapsedTime().TotalSeconds;
48                     // Print the results
49                     Console.WriteLine("Total characters: {0}", totalCharacters);
50                     Console.WriteLine("Total occurrences: {0}", totalOccurrences);
51                     Console.WriteLine("Total time: {0} seconds", totalTime);
52                     Console.WriteLine("Verbose output: {0}", verbose);
53                 }
54             }
55         }
56     }
57 }
```

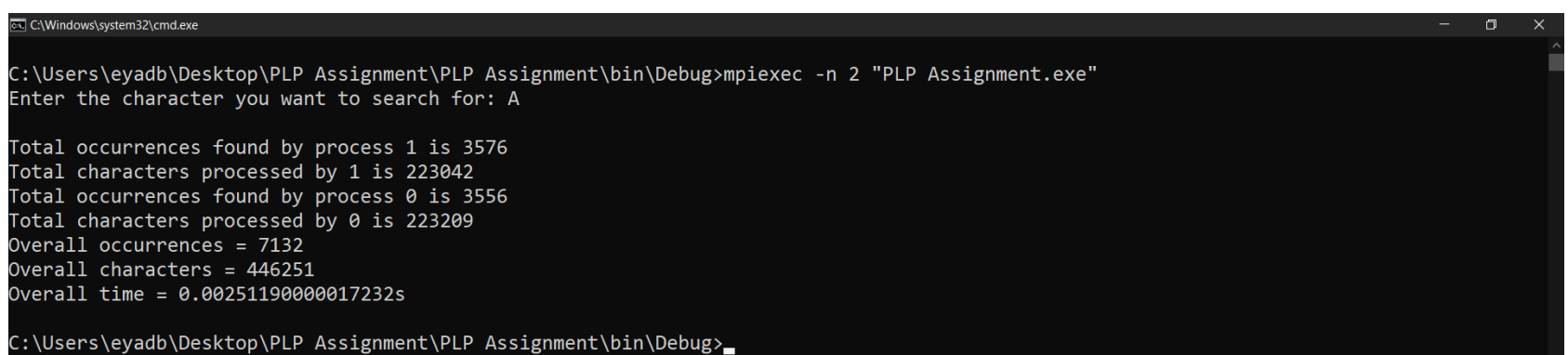
```

38     if (comm.Rank == ROOT_PROCESS_ID)
39     {
40         // Ask the user to enter the character he want to search for
41         Console.Write("Enter the character you want to search for: ");
42         what = (char)Console.Read();
43         contents = File.ReadAllLines(PATH); // Load contents of file inside the array
44     }
45     comm.Broadcast(ref contents, ROOT_PROCESS_ID); // Broadcast contents from root
46     comm.Broadcast(ref what, ROOT_PROCESS_ID); // Broadcast character from root
47
48     // All processes must reach this point before going on
49     // This is important for time measurement
50     comm.Barrier();
51     double start = MPI.Environment.Time; // The start of time measuring
52
53     // Let's divide the load for processes, each process is responsible for a specific number of lines
54     for (int i = comm.Rank; i < contents.Length; i += comm.Size)
55     {
56         if (verbose)
57             Console.WriteLine("Process ID = {0}, Line number = {1}", comm.Rank, i);
58         for (int j = 0; j < contents[i].Length; j++)
59         {
60             if (contents[i][j] == what) // An occurrence has been found
61             {
62                 if (verbose)
63                     Console.WriteLine("Process {0} has found the character at {1}:{2}", comm.Rank, i + 1, j + 1);
64                 totalOccurrences += 1;
65             }
66             totalCharacters += 1;
67         }
68     }
69
70     // All processes must reach here before we stop the timer
71     comm.Barrier();
72     double end = MPI.Environment.Time; // The end of time measurement
73
74     allOccurrences = comm.Reduce(totalOccurrences, Operation.Add, ROOT_PROCESS_ID); // Accumulate all occurrences
75     allCharacters = comm.Reduce(totalCharacters, Operation.Add, ROOT_PROCESS_ID); // Accumulate all characters
76     totalTime = comm.Reduce(end - start, Operation.Max, ROOT_PROCESS_ID); // The overall time is the longest time
77
78     Console.WriteLine("Total occurrences found by process {0} is {1}", comm.Rank, totalOccurrences);
79     Console.WriteLine("Total characters processed by {0} is {1}", comm.Rank, totalCharacters);
80
81     // Finally, if we're here, print all information
82     if (comm.Rank == ROOT_PROCESS_ID)
83     {
84         Console.WriteLine("Overall occurrences = {0}", allOccurrences);
85         Console.WriteLine("Overall characters = {0}", allCharacters);
86         Console.WriteLine("Overall time = {0}s", totalTime);
87     }
88 }
89 return 0;
90 }
91 }
92 }

```

In the very first try, i tried a file with 1000 lines.

Output for **n=2**:



```

C:\Windows\system32\cmd.exe

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpirun -n 2 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 1 is 3576
Total characters processed by 1 is 223042
Total occurrences found by process 0 is 3556
Total characters processed by 0 is 223209
Overall occurrences = 7132
Overall characters = 446251
Overall time = 0.00251190000017232s

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>

```

Detailed output:

```
C:\Windows\system32\cmd.exe
Process 1 has found the character at 998:114
Process 0 has found the character at 999:160
Process 1 has found the character at 998:249
Process 0 has found the character at 999:161
Process 1 has found the character at 998:253
Process 0 has found the character at 999:194
Process 1 has found the character at 998:280
Process 0 has found the character at 999:341
Process 1 has found the character at 998:419
Process 1 has found the character at 998:446
Process 1 has found the character at 998:525
Process 1 has found the character at 998:550
Process ID = 1, Line number = 999
Process 1 has found the character at 1000:37
Process 1 has found the character at 1000:54
Process 1 has found the character at 1000:67
Process 1 has found the character at 1000:86
Process 1 has found the character at 1000:101
Process 1 has found the character at 1000:113
Process 1 has found the character at 1000:165
Process 1 has found the character at 1000:285
Process 1 has found the character at 1000:293
Process 1 has found the character at 1000:394
Process 1 has found the character at 1000:473
Total occurrences found by process 1 is 3576
Total characters processed by 1 is 223042
Total occurrences found by process 0 is 3556
Total characters processed by 0 is 223209
Overall occurrences = 7132
Overall characters = 446251
Overall time = 0.283648700000413s
```

Output for n=4:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpirexec -n 4 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 3 is 1870
Total characters processed by 3 is 112752
Total occurrences found by process 1 is 1706
Total characters processed by 1 is 110290
Total occurrences found by process 0 is 1815
Total characters processed by 0 is 112203
Overall occurrences = 7132
Overall characters = 446251
Overall time = 0.00215529999877617s
Total occurrences found by process 2 is 1741
Total characters processed by 2 is 111006
```

Detailed output:

```
C:\Windows\system32\cmd.exe
Process 3 has found the character at 996:134
Process 3 has found the character at 996:159
Process 3 has found the character at 996:286
Process 3 has found the character at 996:336
Process 3 has found the character at 996:355
Process 3 has found the character at 996:385
Process 3 has found the character at 996:437
Process 3 has found the character at 996:470
Process ID = 3, Line number = 999
Process 3 has found the character at 1000:37
Process 3 has found the character at 1000:54
Process 3 has found the character at 1000:67
Process 3 has found the character at 1000:86
Process 3 has found the character at 1000:101
Process 3 has found the character at 1000:113
Process 3 has found the character at 1000:165
Process 3 has found the character at 1000:285
Process 3 has found the character at 1000:293
Process 3 has found the character at 1000:394
Process 3 has found the character at 1000:473
Total occurrences found by process 2 is 1741
Total occurrences found by process 3 is 1870
Total characters processed by 2 is 111006
Total characters processed by 3 is 112752
Total occurrences found by process 0 is 1815
Total occurrences found by process 1 is 1706
Total characters processed by 0 is 112203
Total characters processed by 1 is 110290
Overall occurrences = 7132
Overall characters = 446251
Overall time = 0.256595999999263s
```

Output for n=8:


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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 8 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 3 is 934
Total characters processed by 3 is 55743
Total occurrences found by process 5 is 852
Total characters processed by 5 is 54616
Total occurrences found by process 1 is 854
Total characters processed by 1 is 55674
Total occurrences found by process 2 is 823
Total characters processed by 2 is 55567
Total occurrences found by process 0 is 907
Total characters processed by 0 is 55737
Overall occurrences = 7132
Overall characters = 446251
Overall time = 0.012319800001139s
Total occurrences found by process 6 is 918
Total characters processed by 6 is 55439
Total occurrences found by process 7 is 936
Total characters processed by 7 is 57009
Total occurrences found by process 4 is 908
Total characters processed by 4 is 56466
```

Detailed output:

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

Process 3 has found the character at 996:286
Process 7 has found the character at 1000:113
Process 3 has found the character at 996:336
Process 7 has found the character at 1000:165
Process 3 has found the character at 996:355
Process 7 has found the character at 1000:285
Process 3 has found the character at 996:385
Process 7 has found the character at 1000:293
Process 3 has found the character at 996:437
Process 7 has found the character at 1000:394
Process 3 has found the character at 996:470
Process 7 has found the character at 1000:473
Total occurrences found by process 1 is 854
Total occurrences found by process 7 is 936
Total occurrences found by process 3 is 934
Total occurrences found by process 2 is 823
Total characters processed by 1 is 55674
Total characters processed by 7 is 57009
Total characters processed by 3 is 55743
Total characters processed by 2 is 55567
Total occurrences found by process 5 is 852
Total characters processed by 5 is 54616
Total occurrences found by process 6 is 918
Total characters processed by 6 is 55439
Total occurrences found by process 0 is 907
Total occurrences found by process 4 is 908
Total characters processed by 0 is 55737
Total characters processed by 4 is 56466
Overall occurrences = 7132
Overall characters = 446251
Overall time = 0.311714399998891s
```

After that, i tried a file with 10000 lines:

Output for **n=2**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>php text-generator.php 10000 300 600 > input.txt

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 2 "PLP Assignment.exe"
Enter the character you want to search for: A
Total occurrences found by process 0 is 36437
Total characters processed by 0 is 2251162
Overall occurrences = 72792
Overall characters = 4504435
Overall time = 0.0250462000003608s
Total occurrences found by process 1 is 36355
Total characters processed by 1 is 2253273
```

Output for **n=4**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 4 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 1 is 18086
Total characters processed by 1 is 1128609
Total occurrences found by process 0 is 18093
Total characters processed by 0 is 1123981
Overall occurrences = 72792
Overall characters = 4504435
Overall time = 0.0177461999992374s
Total occurrences found by process 3 is 18269
Total occurrences found by process 2 is 18344
Total characters processed by 3 is 1124664
Total characters processed by 2 is 1127181
```

Output for **n=8**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 8 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 1 is 8881
Total occurrences found by process 6 is 9221
Total characters processed by 1 is 564170
Total characters processed by 6 is 564626
Total occurrences found by process 7 is 9238
Total characters processed by 7 is 561827
Total occurrences found by process 3 is 9031
Total characters processed by 3 is 562837
Total occurrences found by process 2 is 9123
Total characters processed by 2 is 562555
Total occurrences found by process 5 is 9205
Total characters processed by 5 is 564439
Total occurrences found by process 0 is 9038
Total characters processed by 0 is 557914
Overall occurrences = 72792
Overall characters = 4504435
Overall time = 0.0191967999999179s
Total occurrences found by process 4 is 9055
Total characters processed by 4 is 566067
```

After that, i tried a file with 40000 lines:

Output for **n=2**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 2 "PLP Assignment.exe"
Enter the character you want to search for: A
Total occurrences found by process 0 is 146294
Total characters processed by 0 is 9004321
Overall occurrences = 291097
Overall characters = 17988628
Overall time = 0.1045563999999638s
Total occurrences found by process 1 is 144803
Total characters processed by 1 is 8984307
```

Output for **n=4**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 4 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 1 is 72571
Total characters processed by 1 is 4491902
Total occurrences found by process 3 is 72232
Total characters processed by 3 is 4492405
Total occurrences found by process 2 is 72870
Total characters processed by 2 is 4495053
Total occurrences found by process 0 is 73424
Total characters processed by 0 is 4509268
Overall occurrences = 291097
Overall characters = 17988628
Overall time = 0.06569760000009308s
```

Output for **n=8**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 8 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 1 is 36400
Total characters processed by 1 is 2246623
Total occurrences found by process 5 is 36171
Total characters processed by 5 is 2245279
Total occurrences found by process 3 is 35851
Total characters processed by 3 is 2242092
Total occurrences found by process 2 is 36243
Total characters processed by 2 is 2243153
Total occurrences found by process 0 is 36694
Total characters processed by 0 is 2254937
Overall occurrences = 291097
Overall characters = 17988628
Overall time = 0.05689259999996281s
Total occurrences found by process 7 is 36381
Total occurrences found by process 6 is 36627
Total characters processed by 7 is 2250313
Total characters processed by 6 is 2251900
Total occurrences found by process 4 is 36730
Total characters processed by 4 is 2254331
```

After that, i tried a file with 200000 lines:

Output for **n=2**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 2 "PLP Assignment.exe"
Enter the character you want to search for: A
Total occurrences found by process 0 is 725257
Total characters processed by 0 is 44988899
Overall occurrences = 1451430
Overall characters = 89967075
Overall time = 0.45232500000201s
Total occurrences found by process 1 is 726173
Total characters processed by 1 is 44978176
```

Output for **n=4**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 4 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 1 is 363191
Total characters processed by 1 is 22486703
Total occurrences found by process 3 is 362982
Total characters processed by 3 is 22491473
Total occurrences found by process 0 is 362356
Total characters processed by 0 is 22484216
Overall occurrences = 1451430
Overall characters = 89967075
Overall time = 0.308747600000061s
Total occurrences found by process 2 is 362901
Total characters processed by 2 is 22504683
```

Output for **n=8**:

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

C:\Users\eyadb\Desktop\PLP Assignment\PLP Assignment\bin\Debug>mpiexec -n 8 "PLP Assignment.exe"
Enter the character you want to search for: A

Total occurrences found by process 5 is 181322
Total occurrences found by process 7 is 180946
Total characters processed by 5 is 11232635
Total characters processed by 7 is 11234026
Total occurrences found by process 6 is 181302
Total characters processed by 6 is 11251594
Total occurrences found by process 4 is 181471
Total characters processed by 4 is 11235890
Total occurrences found by process 3 is 182036
Total characters processed by 3 is 11257447
Total occurrences found by process 2 is 181599
Total characters processed by 2 is 11253089
Total occurrences found by process 0 is 180885
Total characters processed by 0 is 11248326
Overall occurrences = 1451430
Overall characters = 89967075
Overall time = 0.239475699998366s
Total occurrences found by process 1 is 181869
Total characters processed by 1 is 11254068
```

For ease, i would like to view the results in a table:

	n = 2	n = 4	n = 8
lines = 10000	0.0250462000003608s	0.0177461999992374s	0.0191967999999179s
lines = 40000	0.104556399999638s	0.0656976000009308s	0.0568925999996281s
lines = 200000	0.452325000000201s	0.308747600000061s	0.239475699998366s

From the table, we can see that there's **approximately** a linear relationship that links between the number of lines and the time, for example:

The required time to process a file that contains 10000 lines using 2 processes equals approximately 25% of the required time to process a file that contains 40000 lines using 2 processes.

However, the relationship between the time and the number of processes isn't linear, but it's sufficient to say that the increment in number of processes yields a smaller execution time.

The time measurement technique depends on the process that takes the longest time.