IVS - profiling

Lidé u výtahu

April 2020

Obsah

1 Úvod **1**

2 Profiling **1**

3 Závěr **1**

4 Přílohy **2**

1 Úvod

Pro výpočet výběrové směrodatné odchylky naše skupina zvolila vývoj Konzolové aplikace s názvem SampleStandardDeviation.exe. Tato aplikace používá třídu IVSMath s matematickými operacemi. Ze standardního vstupu načte libovolný počet čísel a na standardní výstup vypíše výběrovou směrodatnou odchylku.

2 Profiling

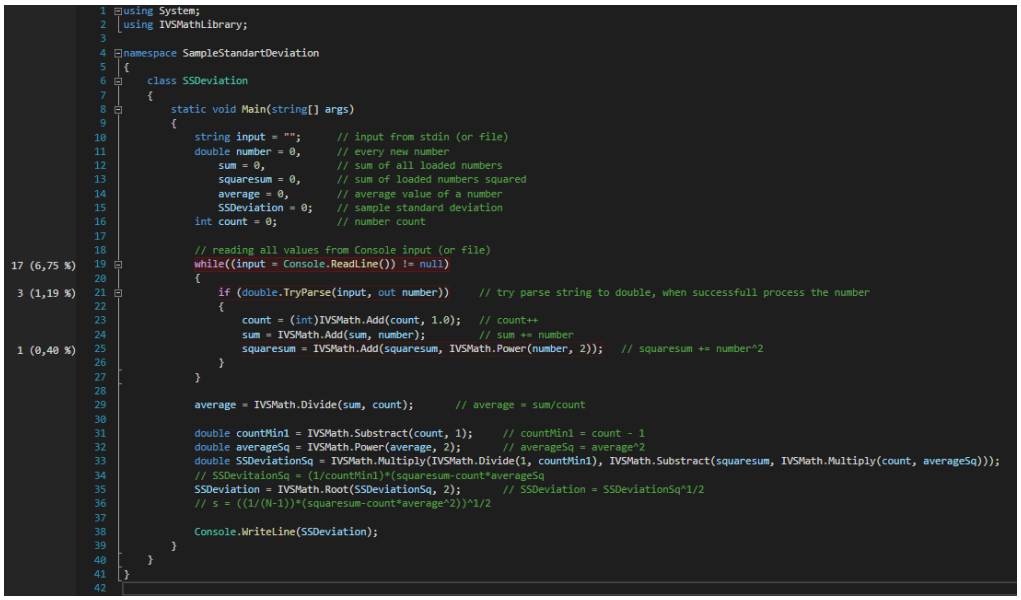
Ve Visual Studiu 2019 jsme naši aplikaci profilovali pomocí Performance profileru, který jako výstup vytváří soubor s příponou .diagsession. Tento soubor obsahuje veškerá data zjištěná při profilování a dá se otevřít přímo ve Visual Studiu.

Mimo jiné obsahuje tabulku funkcí s jednotkami CPU, které daná funkce spotřebovala za běhu aplikace. Zobrazuje funkce, které spotřebují alespoň jednu jednotku CPU, ostatní nezahrnuje. To stejné platí pro zobrazení náročnosti jednotlivých řádků. Na obrázku 1 je vidět příklad výstupu profileru pro vstup s 1000 čísel. Zobrazuje řádky, které zabírají v průběhu programu nejvíce CPU.

3 Závěr

Tato metoda profilování přinesla jasnou představu o tom, které řádky kódu jsou prováděné nejčastěji a zaberou nejvíce času na CPU. Při případné optimalizaci

bychom se měli soustředit právě na tyto klíčové oblasti.



```
                1  using System;
                2  using IVSMathLibrary;
                3
                4  namespace SampleStandartDeviation
                5  {
                6      class SSDeviation
                7      {
                8          static void Main(string[] args)
                9          {
               10              string input = "";      // input from stdin (or file)
               11              double number = 0,      // every new number
               12                  sum = 0,            // sum of all loaded numbers
               13                  squaresum = 0,      // sum of loaded numbers squared
               14                  average = 0,        // average value of a number
               15                  SSDeviation = 0;    // sample standard deviation
               16              int count = 0;          // number count
               17
               18              // reading all values from Console input (or file)
17 (6,75 %)    19              while((input = Console.ReadLine()) != null)
               20              {
 3 (1,19 %)    21                  if (double.TryParse(input, out number))     // try parse string to double, when successfull process the number
               22                  {
               23                      count = (int)IVSMath.Add(count, 1.0);   // count++
               24                      sum = IVSMath.Add(sum, number);         // sum += number
 1 (0,40 %)    25                      squaresum = IVSMath.Add(squaresum, IVSMath.Power(number, 2));   // squaresum += number^2
               26                  }
               27              }
               28
               29              average = IVSMath.Divide(sum, count);       // average = sum/count
               30
               31              double countMin1 = IVSMath.Substract(count, 1);     // countMin1 = count - 1
               32              double averageSq = IVSMath.Power(average, 2);       // averageSq = average^2
               33              double SSDeviationSq = IVSMath.Multiply(IVSMath.Divide(1, countMin1), IVSMath.Substract(squaresum, IVSMath.Multiply(count, averageSq)));
               34              // SSDevitaionSq = (1/countMin1)*(squaresum-count*averageSq
               35              SSDeviation = IVSMath.Root(SSDeviationSq, 2);       // SSDeviation = SSDeviationSq^1/2
               36              // s = ((1/(N-1))*(squaresum-count*average^2))^1/2
               37
               38              Console.WriteLine(SSDeviation);
               39          }
               40      }
               41  }
               42
```

Obrázek 1: Výstup profileru

4 Přílohy

Název souboru	Popis
vystup-data10.diagsession	výstup profilování s 10 vstupy
vystup-data10.png	screenshot využití kódu s 10 vstupy
vystup-data100.diagsession	výstup profilování se 100 vstupy
vystup-data10.png	screenshot využití kódu se 100 vstupy
vystup-data1000.diagsession	výstup profilování s 1000 vstupů
vystup-data10.png	screenshot využití kódu s 1000 vstupů