Michał Szczygieł	Aleksander Śmierciak	18/06/2014

Zadanie 4 - Trigramy - Wykrywanie języka

Programem spełniającym polecenie zadania 4. jest taki, który przyjmując plik wejściowy uzyskany z wykonania programu z zadania 3. przeprowadzi dedukcję języka, w którym tekst z pliku został zapisany.

Część programu odpowiedzialna za zrównoleglenie:

```
1       [...]
2         #pragma omp parallel for num_threads(threadCount) \
3                     shared(trigramsToCompare, result) \
4                     firstprivate(files) schedule(dynamic)
5             for (unsigned int i = 0; i < files->size(); ++i)
6             {
7                     result[files->at(i)] = compareTrigrams(
8                         trigrams, trigramsToCompare[files->at(i)]);
9             }
10    [...]
11
12 double compareTrigrams(Histogram trigrams, Histogram trigramsToCompare)
13 {
14     int maxCoverage = trigrams.size();
15     int currentCoverage = 0;
16     for (Histogram::iterator iti = trigrams.begin();
17                              iti != trigrams.end();
18                            ++iti)
19     {
20         for (Histogram::iterator itj = trigramsToCompare.begin();
21                                  itj != trigramsToCompare.end();
22                                ++itj)
23         {
24             if((*iti).first == (*itj).first)
25             {
26                 currentCoverage++;
27             }
28         }
29     }
30     return (100 * currentCoverage) / (double) maxCoverage;
31 }
```

Wcześniejsza implementacja:

```
1    #pragma omp parallel num_threads(threadCount)
2    {
3       std::string threeLetters;
4       unsigned int endPosition = portion * (omp_get_thread_num() + 1);
5
6       if (endPosition > contents.size())
7       {
8          endPosition = contents.size();
9       }
10
11      #pragma for default(none) shared(contents, trigram) firstprivate(
            portion) private(threeLetters)
12      for (int i = portion * omp_get_thread_num();
```

```
13              i != portion * (omp_get_thread_num() + 1); i += 3)
14      {
15          threeLetters = std::string(contents.substr(i, 3));
16          trigram[threeLetters]++;
17      }
18  }
```

Przebieg

Obliczenia zostały wykonane na serwerze CUDA. Oprócz tego zostały zastosowane dwie równoległe sekcje; najpierw plik podlega analizie, a następnie przeprowadza dedukcję. W tym programie również zaimplementowane zostały mechanizmy zamków (tylko do analizy), natomiast w cześci rozpoznawania, obeszło się bez machanizmów zamków, gdyż iteratory zapewniają bezpieczny dostęp.

Poniżej wykresy przedstawiający rezultat przeprowadzonych operacji na wątkach.