

Cel zadania

Celem zadania jest głębsze zapoznanie ze środowiskiem obliczeń rozproszonych OpenMP. Ćwiczenie wykorzystuje wiedzę z przedmiotu statystyka.

Zadanie 3 - Trigramy - Analiza dokumentów

Napisz program do analizy statystycznej dokumentów. Program odczytuje pliki tekstowe i odczytany tekst dzieli na trigramy (trzyliterowe podciągi), a następnie częstotliwość ich występowania zapisuje w pliku wynikowym o nazwie <kod języka>.dat. Program należy wykonać z użyciem OpenMP.

Przykład dla języka polskiego:

MARCELEK MARZY O AEROPLANIE STEROWANYM ZDALNIE

należy podzielić na

'MAR' 'CEL' 'EK' 'MAR' 'ZY' 'O A' 'ERO' 'PLA'
'NIE' 'ST' 'ERO' 'WAN' 'YM' 'ZDA' 'LNI'

do pliku wynikowego pl.dat należy wpisać:

MAR 2
CEL 1
EK 1
ZY 1
O A 1
ERO 2
PLA 1
NIE 1
ST 1
WAN 1
YM 1
ZDA 1
LNI 1

Jako pliki wejściowe można podać słowniki z katalogu /usr/share/dict/

Wymagania

Program powinien:

- mierzyć łączny czas dla wszystkich podanych plików łącznie,
- mierzyć tylko czas obliczeń bez wypisywania tekstu oraz odczytu/zapisu plików,
- wypisać czas obliczeń podany w ms.

Dokumentacja

Dokumentacja oprócz standardowych elementów powinna jeszcze zawierać wykresy zależności czasu obliczeń od ilości wątków oraz wykres przyspieszenia. Mierzony czas powinien być nie mniejszy niż 3 s.

Położenie plików

- Program: `./zad3/analiza_omp`
- Dokumentacja L^AT_EX: `./zad3/dok.tex`
- Dokumentacja PDF: `./zad3/dok.pdf`

Uruchamianie

```
./analiza_omp <count> <lang> <file1> [<file2> ...]
```

- `count` - liczba procesów/wątków,
- `lang` - kod analizowanego języka
- `fileN` - ścieżki analizowanych plików

Zadanie 4 - Trigramy - Wykrywanie języka

Napisz program do wykrywania języka na podstawie utworzonych w poprzednim zadaniu plików z analizą dla danego języka. Analizuje plik budując trigramy i porównuje z trigramami dla danych języków. Program należy wykonać z użyciem OpenMP.

Wymagania

Program powinien:

- wypisać języki w kolejności od najbardziej prawdopodobnego do najmniej z określeniem procentowym dopasowania,
- mierzyć tylko czas obliczeń bez wypisywania tekstu oraz odczytu plików,
- wypisać czas obliczeń podany w ms.

Dokumentacja

Dokumentacja oprócz standardowych elementów powinna jeszcze zawierać wykresy zależności czasu obliczeń od ilości wątków oraz wykres przyspieszenia. Mierzony czas powinien być nie mniejszy niż 3 s.

Położenie plików

- Program: `./zad4/wykrywanie_omp`
- Dokumentacja L^AT_EX: `./zad4/dok.tex`
- Dokumentacja PDF: `./zad4/dok.pdf`

Uruchamianie

```
./wykrywanie_omp <count> <file>
```

- `count` - liczba procesów/wątków,
- `file` - ścieżka badanego pliku

Zadanie 5 - Mnożenie macierzy na GPU

Napisz program, który oblicza iloczyn macierzy z użyciem karty graficznej w technologii CUDA, ATI Stream lub OpenCL.

Wymagania

Program powinien:

- mierzyć tylko czas obliczeń bez wypisywania tekstu i alokacji pamięci, ani kopiowania pamięci pomiędzy urządzeniami,
- wypisać czas obliczeń podany w ms,
- mierzyć czas dla liczby wątków która jest wielokrotnością liczby 2.

Dokumentacja

Dokumentacja oprócz standardowych elementów powinna jeszcze zawierać wykresy zależności czasu obliczeń od ilości wątków (w tym liczby rzeczywistych procesorów karty graficznej) oraz wykres przyspieszenia. Mierzony czas powinien być nie mniejszy niż 3 s.

Położenie plików

- Program: `./zad5/macierz_cuda`
- Dokumentacja L^AT_EX: `./zad5/dok.tex`
- Dokumentacja PDF: `./zad5/dok.pdf`

Uruchamianie

```
./macierz_cuda <count> <size>
```

- `count` - liczba wątków,
- `size` - rozmiar macierzy