**SPRAWOZDANIE**

Przedmiot: Programowanie komputerów wielordzeniowych

Laboratorium: 2

Student: Artur Ziemba

Grupa: C

Celem sprawozdania było sprawdzenie czasu pracy programu wykonującego operacje mnożenia macierzy w zależności od ilości wykorzystanych wątków systemu operacyjnego przy użyciu języka c lub c++ oraz biblioteki Openmp.

Wyniki zostały uzyskane na komputerze z systemem Windows 7 (64bit), 16GB RAM oraz procesorze Intel(R) Core(TM) i7-7700K CPU 4.20 GHz, 4 korowym, (8 procesorów logicznych). Do kompilacji programu wykorzystano kompilator g++. Każdy test został powtórzony 10 razy, a uzyskane wyniki prezentują wartość średnią.

1. Implementacja macierzy przy użyciu dynamicznej tablicy dwuwymiarowej:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | iterowanie przez kolumny, następnie przez wiersze | | iterowanie przez wiersze, następnie przez kolumny | | Algorytm Strassena | |
| rozmiary macierzy / ilość wątków | 50x50 | 500x500 | 50x50 | 500x500 | 50x50 | 500x500 |
| 1 | 0.006296 | 7.50796 | 0.006241 | 7.29681 | 0.006121 | 7.084263 |
| 2 | 0.003393 | 4.069986 | 0.003719 | 4.029703 | 0.003621 | 3.783552 |
| 3 | 0.003194 | 3.754953 | 0.003417 | 3.791901 | 0.003047 | 3.792550 |
| 4 | 0.003130 | 3.697121 | 0.003136 | 3.390330 | 0.003096 | 3.465918 |
| 5 | 0.002708 | 3.440928 | 0.002883 | 3.551365 | 0.002852 | 2.884350 |
| 6 | 0.003111 | 3.482078 | 0.002969 | 3.766306 | 0.003037 | 3.334519 |
| 7 | 0.002792 | 2.819516 | 0.002428 | 3.061478 | 0.002201 | 2.577736 |
| 8 | 0.002475 | 2.436026 | 0.002033 | 2.820948 | 0.001953 | 2.786490 |

1. Implementacja macierzy na kolekcji std::vector:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | iterowanie przez kolumny, następnie przez wiersze | | iterowanie przez wiersze, następnie przez kolumny | | Algorytm Strassena | |
| rozmiary macierzy / ilość wątków | 50x50 | 500x500 | 50x50 | 500x500 | 50x50 | 500x500 |
| 1 | 0.104251 | 104.447 | 0.106069 | 104.506 | 0.094208 | 90.836384 |
| 2 | 0.057465 | 56.093306 | 0.060235 | 56.661108 | 0.056274 | 47.322162 |
| 3 | 0.058696 | 53.929176 | 0.054060 | 56.442598 | 0.048958 | 46.943657 |
| 4 | 0.055166 | 52.200395 | 0.055101 | 49.144656 | 0.044343 | 47.267511 |
| 5 | 0.050729 | 49.427870 | 0.050436 | 44.158277 | 0.039846 | 35.535192 |
| 6 | 0.046536 | 47.871810 | 0.049019 | 55.408548 | 0.047826 | 46.657464 |
| 7 | 0.041971 | 40.003193 | 0.038472 | 46.163950 | 0.033557 | 39.612378 |
| 8 | 0.038500 | 39.726641 | 0.032626 | 38.276487 | 0.029618 | 35.207560 |

Duża wartość czasowa uzyskana dla macierzy 500x500 dla tej kolekcji jest spowodowana sposobem zapisu kolekcji w pamięci (musi zostać dopasowana do ciągłego obszaru pamięci). W związku z tym znaczący okres pracy zajęło alokowanie zasobów.

1. Podsumowanie wyników:

Uzyskane wyniki pokazują, że największa sprawność została uzyskana dla rozwiązania bazującego na tablicy dwuwymiarowej oraz algorytmie Strassena. Względem programu jednowątkowego dla macierzy 50x50 najlepszą sprawność uzyskano na ośmiu wątkach - przyśpieszenie o 0.004168 sekundy (68%), natomiast dla macierzy 500x500 najlepszą sprawność uzyskano na siedmiu wątkach – przyśpieszenie o 4,506527 sekundy (64%).