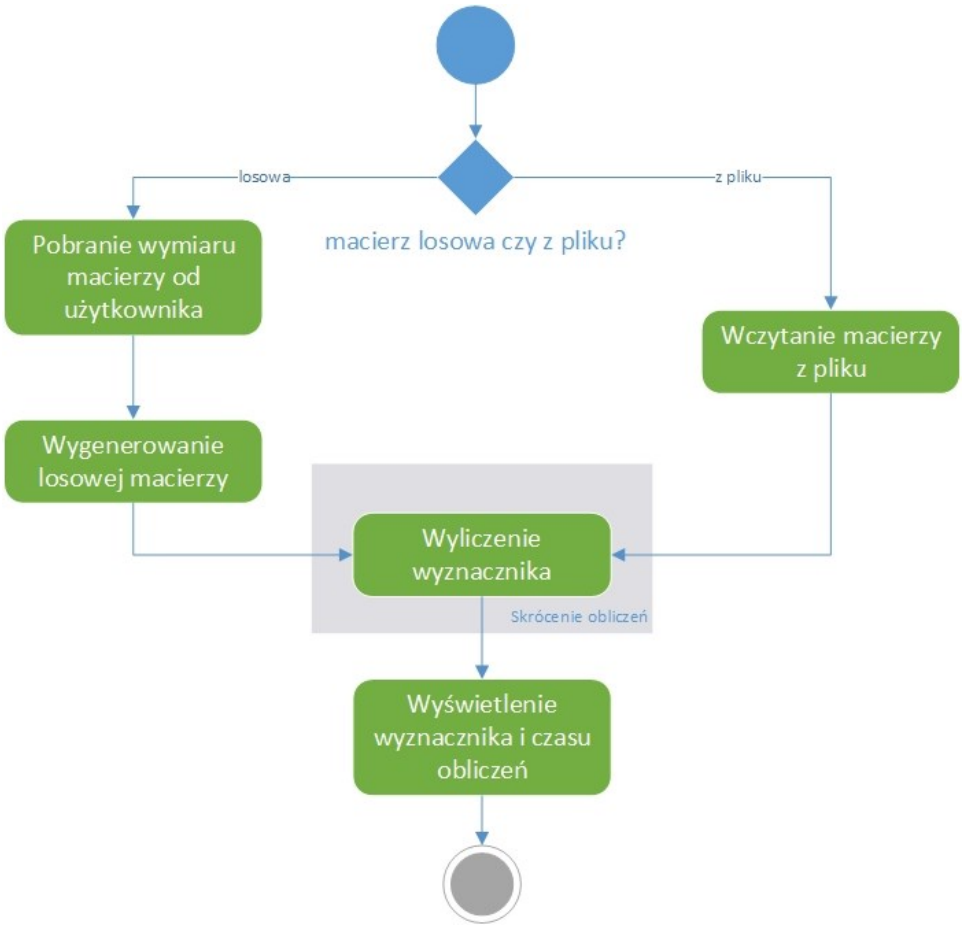
Laboratorium Grafiki 3D i Wizualizacji Komputerowej		
Projekt III – Wyznacznik macierzy		
Krzysztof Krawulski Krzysztof Sommerrey	2016-01-17	AiR mgr. Sem. II rok II Grupa KSS

1. Cel projektu

Celem projektu jest napisanie algorytmu obliczania wyznacznika macierzy o dowolnym rozmiarze (macierz musi być kwadratowa) i wykorzystanie biblioteki OpenMP do przyspieszenia obliczeń.

2. Diagram UML i opis programu

Prosty diagram UML przedstawiony został na grafice 2.1.



Grafika 2.1 Diagram UML

Program został napisany w jednym pliku o nazwie *main.cpp*. W kodzie można wyróżnić 6 funkcji:

- void LosowanieMacierzy() – funkcja odpowiedzialna za utworzenie macierzy o podanym przez użytkownika wymiarze oraz wypełnienia jej cyframi losowanymi z przedziału od zera do dziewięciu. Funkcja ta była również wykorzystywana do zapisania jednej losowej macierzy do pliku tekstowego (odpowiedni kod został zakomentowany);
- void OdczytMacierzy() – tutaj zostaje odczytana macierz z pliku tekstowego – jeżeli użytkownik wybierze taką opcję. Początkowo funkcja ta miała zostać użyta do pomiarów czasu obliczeń;
- void Obliczenia() – funkcja, gdzie mają miejsce wszystkie obliczenia potrzebne do uzyskania wyznacznika danej macierzy. Algorytm wykorzystany do obliczeń bazuje na metodzie eleminacji Gaussa;
- void ObliczeniaOpenMP() – te same obliczenia, jednak z wykorzystaniem przyspieszenia sprzętowego;
- void PomiarCzasu() – tutaj wywoływane są funkcje obliczeń w pętlach *for*, każda 1000 razy. Pozwala to osiągnąć czasy łatwiejsze do późniejszej analizy;
- int main() – główna funkcja programu – tutaj zapisane jest menu wyświetlające się użytkownikowi oraz następują wywołania powyższych funkcji.

Macierz zapisana do pliku ma wymiar 100x100. Dzięki temu wydłuża się czas obliczeń dla komputera. Należy jednak pamiętać, że wyznaczniki tak dużych macierzy osiągają bardzo dużą wartość bezwzględną.

3. Opis sposobu zrównoleglenia

Do zrównoleglenia obliczeń zostały użyte dwie dyrektywy:

- #pragma omp parallel for default(shared) private(i,j);
- #pragma omp parallel for default(none) shared (N, tab) private(i) \
  reduction(* : detMP).

Obie dyrektywy odnoszą się do pętli *for*. Pierwsza z nich zmienne *i* oraz *j* bierze jako prywatne. Oznacza to, że biblioteka *OpenMP* może je swobodnie rozdzielić między dostępne wątki. Przykładowo, jeżeli mamy pętlę *for* ze zmienną *i* w zakresie od 0 do 1000 oraz czterowątkowy procesor, to do każdego z wątków zostanie przydzielone po 250 iteracji.

Wszystkie inne zmienne w danej pętli są oznaczone jako domyślnie współdzielone. Oznacza to, że każdy z wątków będzie je przeliczał od nowa (nie są rozdzielone jak w *i* w powyższym przykładzie). Do takich zmiennych można zaliczyć na przykład zmienną, gdzie zapisywany jest wynik obliczeń wykonywanych w pętli.

Druga dyrektywa ma konkretnie określone, które zmienne są współdzielone, a które prywatne dla każdego z wątków. Procesory będą się dzielić zmiennymi *N* oraz *tab*, natomiast prywatną zmienną

jest *i*. Dodatkowo w tej dyrektywnie można zauważyć, że wykorzystano *reduction*. Informuje to kompilator, że do zmiennej *detMP* wpisywany będzie wynik jej mnożenia z konkretnym elementem tablicy (*detMP = detMP* tab[i][i]*). Utworzone zostaną prywatne kopie zmiennej dla każdego z wątków i rozdzielone iteracje. Po wykonaniu wszystkich iteracji wartości wyliczone przez wszystkie wątki zostaną przemnożone przez siebie.

4. Analiza zależności między danymi

Wykorzystanie metody Gaussa do obliczenia wyznacznika macierzy powoduje, że obliczane dane nie zależą od kolejnych iteracji, gdyż operacja liczenia wartości wyznacznika jest wykonywana po obliczeniu wartość komórek macierzy na diagonali. Różnice w czasach wykonywania będą wynikać głównie z faktu, iż w zależności od zestawu danych wejściowych, który jest generowany losowo, zmienia się złożoność obliczeniowa. Zgodnie z prawem Amdahla, pozwalającym obliczyć maksymalny przyrost wydajności obliczeniowej uzyskiwanej poprzez zrównoleglenie obliczeń, możemy obliczyć przewidywany przyrost. Ze wzoru:

$$\frac{1}{(1-P)+\frac{P}{N}}$$

gdzie P oznacza procent zrównoleglanych obliczeń (przyjmujemy 90%), a N oznacza liczbę procesorów (ze względu na wielowątkowość, możemy traktować testowane procesory jako 4 i 8 rdzeniowe) otrzymujemy przewidywane przyspieszenie:

Dla czterowątkowego procesora: $\frac{1}{(1-0,9)+\frac{0,9}{4}} = 3,077$

Dla ośmiowątkowego procesora: $\frac{1}{(1-0,9)+\frac{0,9}{8}} = 4,71$

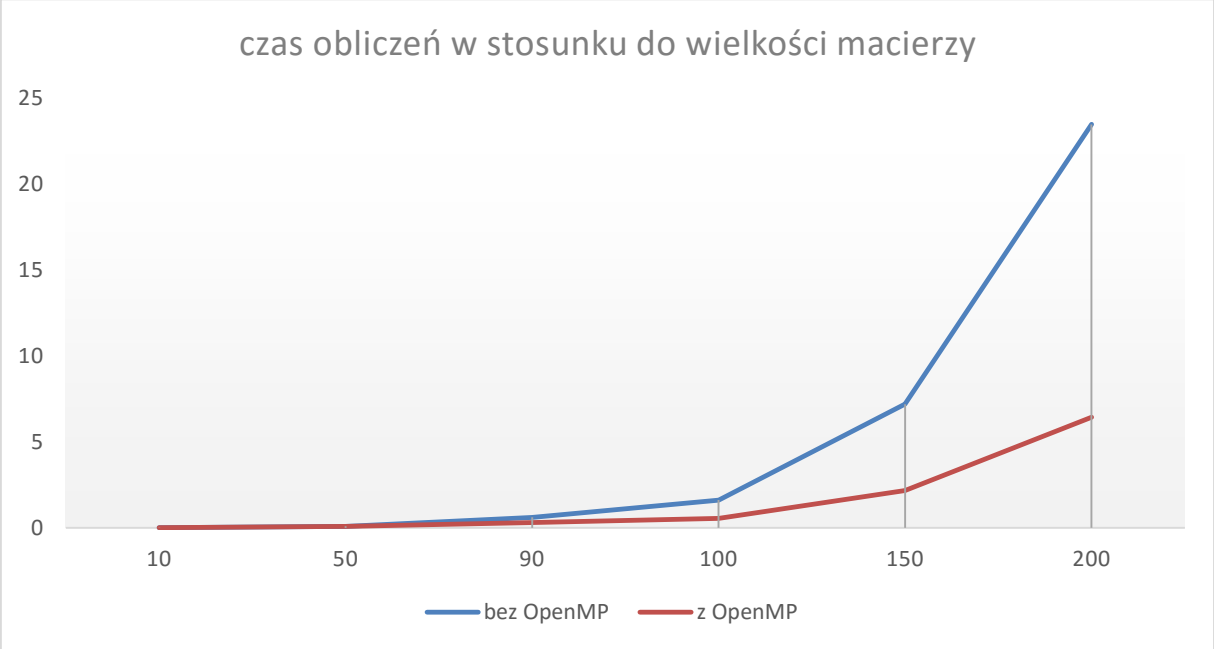
5. Obliczenia i wykresy

Obliczenia czasów robione były na dwóch komputerach. Pierwszy z nich dysponuje procesorem *Intel Core i7 – 4700QM* (4 rdzenie, 8 wątków), drugi procesorem Intel Core i3 – 350M (2 rdzenie, 4 wątki). Zrobionych zostało po pięć pomiarów dla macierzy o N = 10, 50, 90, 100, 150 i 200 (wymiar macierzy to NxN).

Wyniki dla pierwszego z komputerów – macierz była losowana za każdym razem:

i\t	bez OpenMP [s]					z OpenMP (8 wątków) [s]				
10	0.000971	0.001059	0.00097	0.001064	0.000729	0.014804	0.009744	0.009816	0.014888	0.016337
50	0.087749	0.088191	0.087801	0.08765	0.089932	0.063051	0.084883	0.073795	0.090339	0.088277
90	0.798703	0.514019	0.658549	0.529635	0.527813	0.450487	0.242278	0.318653	0.239791	0.240584
100	1.27644	0.731565	0.727342	4.2054	1.11069	0.584064	0.305138	0.347102	1.027619	0.44352
150	14.8966	12.6316	2.93182	3.03599	2.47837	4.07673	4.24411	0.793575	0.814107	0.861596
200	18.9928	44.5999	6.17778	41.2854	6.32183	4.85815	11.1568	3.46465	10.5765	2.07843

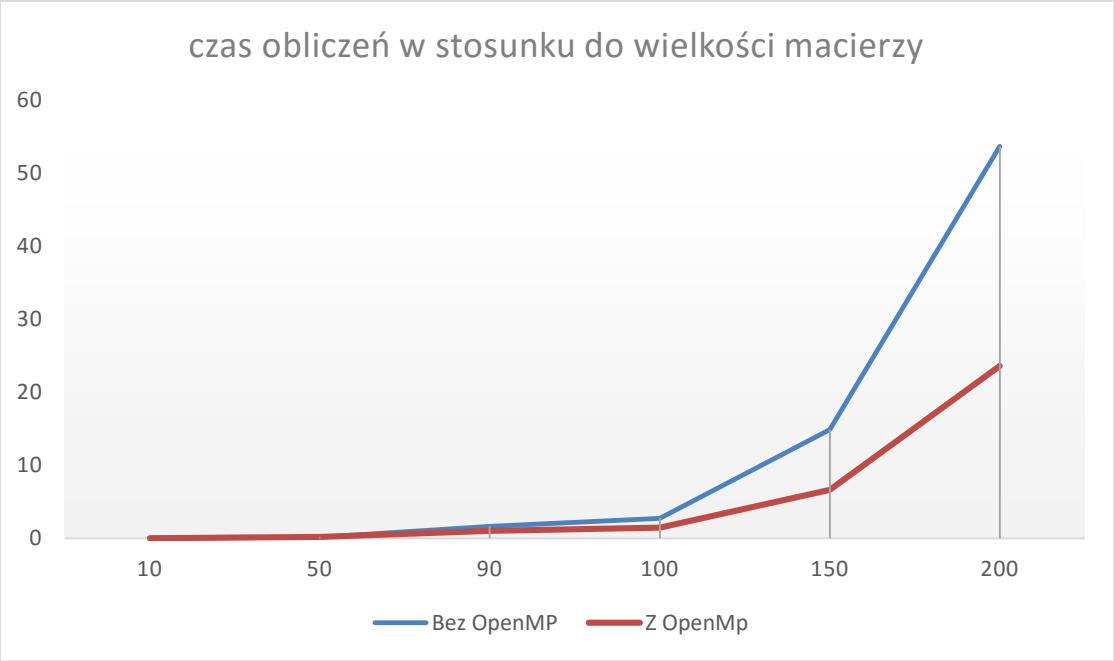
Poniższy wykres bazuje na uśrednionych wynikach ze wszystkich pięciu prób pomiarowych:



Jak można wywnioskować z wykresu różnica w czasach obliczeń zaczyna się dla macierzy o wielkości powyżej 50x50. Im większy wymiar macierzy tym większa robi się różnica w czasach obliczeń dla obu metod.

Wyniki dla drugiego z komputerów – macierz była losowana za każdym razem:

i\t	bez OpenMP [s]					z OpenMP (4 wątki) [s]				
10	0.0025547	0.0027866	0.002786	0.003239	0.002451	0.028363	0.034450	0.033910	0.027996	0.034736
50	0.161282	0.161487	0.168265	0.136846	0.156094	0.16581	0.168484	0.160406	0.151716	0.241272
90	2.05356	2.53316	1.13674	0.869397	1.6167	1.29731	1.40049	0.711491	0.703448	1.1206
100	3.45612	4.81481	1.27458	1.56191	2.6046	1.76153	2.19283	0.89779	0.889961	1.64174
150	11.5262	3.62519	5.83873	33.2382	20.2039	4.83358	2.1719	2.95696	14.0089	9.25773
200	80.9675	72.7503	8.95184	9.01922	96.5661	38.2057	30.9425	5.50829	5.8898	37.3949



Można zauważyć, iż pomimo wydłużenia się średniego czasu obliczeń dla większych rozmiarów macierzy, związanego z użyciem mniejszej liczby wątków, kształt przebiegów jest bardzo zbliżony do uzyskanych przez pierwszy komputer. Ponownie możemy zaobserwować, że rozdział obliczeń na poszczególne wątki jest opłacalny dla rozmiarów macierzy powyżej 50x50, gdyż dla mniejszych wartości, sam rozdział zadań zajmuje więcej czasu, niż ich wykonanie.

6. Wnioski i uwagi

Wykonanie serii testów programu obliczającego wyznacznik macierzy pokazało nam, iż dla wraz ze wzrostem liczby wykonywanych przez procesor obliczeń, wykorzystanie dyrektyw OpenMP umożliwia znacznie efektywniejsze wykorzystanie zasobów procesora, co umożliwia skrócenie czasu wykonania zadania. Warto jednak zaznaczyć, że czas obliczeń stawał się krótszy jedynie w przypadku, gdy obliczany był wyznacznik macierzy o rozmiarach większych niż 50x50, a dla niższych wartości mógł się nawet wydłużać. Jest to najprawdopodobniej związane z faktem, iż czas na wykonanie dyrektyw OpenMP i podział zadania na wątki był dłuższy niż sam czas wykonania obliczeń. Testy zostały przeprowadzone dla dwóch różnych procesorów (ośmio- i czterowątkowego) pokazując w przybliżeniu odpowiednio czteroipółkrotne (4,47) oraz trzykrotne (2,98) przyspieszenie obliczeń dla maksymalnego sprawdzanego rozmiaru macierzy, co pokrywa się w sposób satysfakcjonujący z przewidywanymi wartościami wynikającymi z prawa Amdahla.