

## Zadanie 1 - Całkowanie OpenMP

Ćwiczenie polegało na policzeniu całki oznaczonej metodą trapezów korzystając z biblioteki OpenMP w celu zrównoleglenia obliczeń. Aby móc przekazać do programu wzór funkcji jako parametr wywołania wykorzystano bibliotekę ExprTk. Poniżej znajduje się fragment kodu, który ulega zrównolegleniu:

```
1 #pragma omp parallel num_threads(nThreads) private(x,tmp) shared(from,to,
   nSteps)
2 {
3     IParser* parser = parserFactory->Build();
4 #pragma omp for reduction(+:result)
5     for (int i = 1; i < nSteps - 1; i++)
6     {
7         x = from + fabs(from - to) * (i / (double)nSteps);
8         tmp = parser->Eval(x);
9         result += tmp;
10    }
11 #pragma omp single
12    {
13        result += (parser->Eval(from) + parser->Eval(to)) / 2;
14    }
15 }
16 return result*step;
```

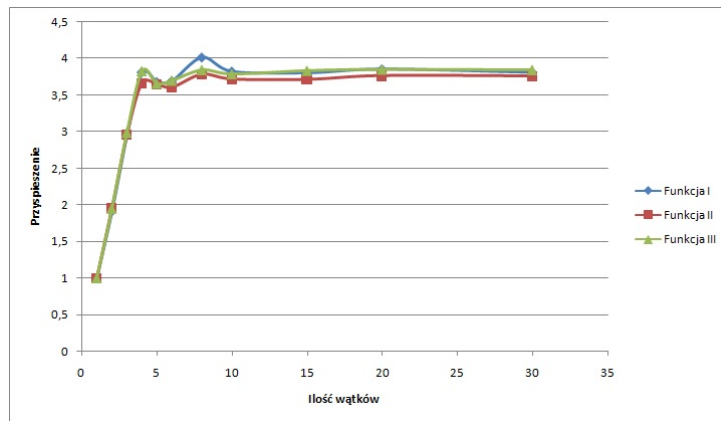
Przy pomocy dyrektywy parallel przydzielono daną liczbę wątków do procesu zrównoleglenia. Parametr funkcji wraz z jej tymczasową wartością są prywatne dla każdego wątku, zaś przedział całkowania z liczbą trapezów są współdzielone. W celu zwiększenia wydajności przekształcono wzór liczący sumę pól trapezów umieszczając jego część w dyrektywie reduction umieszczając wartość sumy ze wszystkich wątków do jednej zmiennej globalnej oraz w dyrektywie single, która była liczona przez tylko jeden wątek.

Zebrano wyniki z trzech różnych funkcji oraz zestawiono poniżej w postaci dwóch wykresów - zależności przyspieszenia obliczeń oraz ich czasu od liczby wątków. Ich wzory znajdują się poniżej:

$$I : f(x) = x^2 + 7x + 2$$

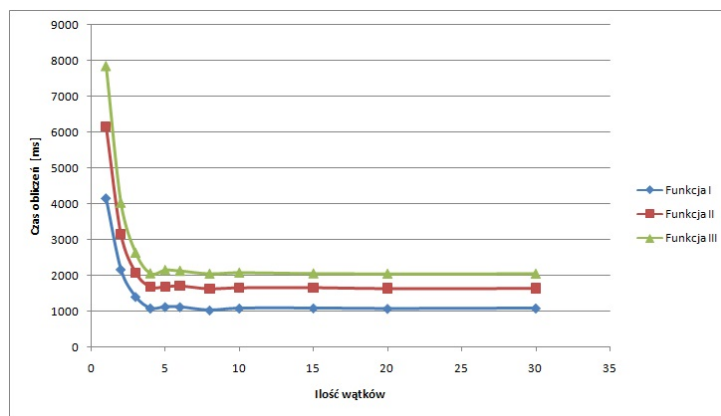
$$II : f(x) = x^3 + 4x + 13$$

$$III : f(x) = 4x^3 + x^2 + 8x$$



Rysunek 1: Wykres zależności przyspieszenia obliczeń od liczby wątków

Patrząc na powyższy wykres można zauważyć kilka zależności. Wszystkie trzy funkcje przyspieszają w niemal identyczny sposób. Przy użyciu 2-4 wątków wartość przyspieszenia była bliska idealnej, z największą zanotowaną stratą przy czterech. Lekki spadek wydajności zanotowano dla 5-7 wątków wracając do wcześniejszego stanu przy 8 i kolejnych wątkach. Zjawiska te mają miejsce z powodu narzutu tworzenia kolejnych wątków, przez co maksimum wydajności można zaobserwować dla 4 lub 8 wątków.



Rysunek 2: Wykres zależności czasu obliczeń od liczby wątków

Podobne wnioski można wysnuć obserwując drugi wykres pokazujący, jak zmienia się czas obliczeń do kolejnych wątków. Każda kolejna funkcja jest bardziej skomplikowana od poprzedniej, co wymaga więcej czasu na dokonanie obliczeń, jednakże sposób, w jaki zachowuje się wydajność jest niemal identyczny.

Wnioski z wykonanego ćwiczenia:

- Biblioteka OpenMP jest bardzo dobrym narzędziem do tworzenia programów równoległych.
- Otrzymane wyniki przyspieszeń są zgodne z oczekiwanymi - dla 2,3 i 4 wątków następuje wzrost wydajności, następnie występuje jej lekki spadek, po czym utrzymuje stały poziom.
- Poziom skomplikowania funkcji wpływa na czas jej obliczeń.