

Łukasz Oprych Gr. Lab. 5 Informatyka Techniczna	Wydajność	13.01.2024
---	-----------	------------

Cel ćwiczenia:

Doskonalenie umiejętności analizy wydajności programów równoległych.

Przebieg ćwiczenia:

Po przygotowaniu struktury katalogowej i skopiowaniu pliku ze strony prowadzącego, skompilowano program `calka_omp.c` i wywołano program za pomocą polecenia:

`export OMP_NUM_THREADS=il. Wątków && ./calka_omp`

Przykładowy wynik dla pomiaru dla 4 wątków:

```

Czas wykonania 0.000344. Obliczona całka = 0.2499999999999991
loprych@loprych-VirtualBox:~/Desktop/PR_lab/lab_13/calca$ export OMP_NUM_THREADS
=4 && ./calca_omp

Program obliczania całki metodą trapezów.

Początek obliczeń OpenMP
Czas wykonania 0.036860. Obliczona całka = 0.2499999999999991

```

Wszystkie wyniki zestawiono w pliku excel:

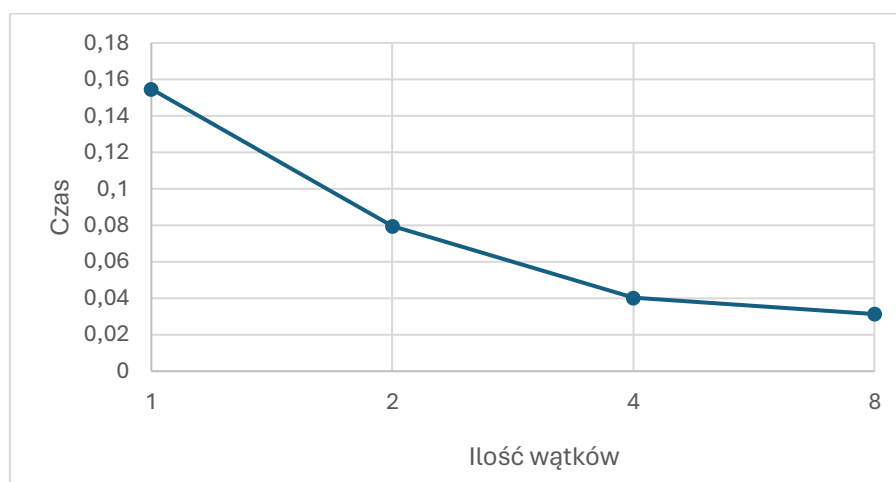
Nr pomiaru		Num_threads = 1	
1	0,150627		
2	0,138869		
3	0,15886		
Średni czas	0,154744		
Nr pomiaru		Num_threads = 2	
1	0,078664		
2	0,078634		
3	0,080306		
Średni czas	0,079485		
Nr pomiaru		Num_threads = 4	
1	0,041242		
2	0,040289		
3	0,039306		
Średni czas	0,040274		
Nr pomiaru		Num_threads = 8	
1	0,028904		
2	0,031024		
3	0,033802		
Średni czas	0,031353		

Następnie dokonano obliczeń dla efektywności i przyspieszenia względnego:

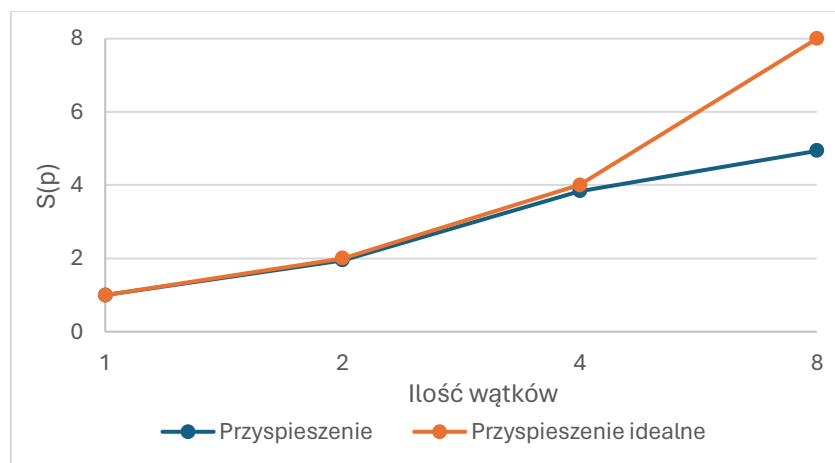
L. wątków	Czas	Przyspieszenie	Efektywno	Przyspiesz.	Efektywność idealna
1	0,154744	1	1	1	1
2	0,079485	1,946826445	0,973413	2	1
4	0,040274	3,842267964	0,960567	4	1
8	0,031353	4,935524511	0,616941	8	1

Następnie zestawiono wykresy zależności od liczby wątków dla:

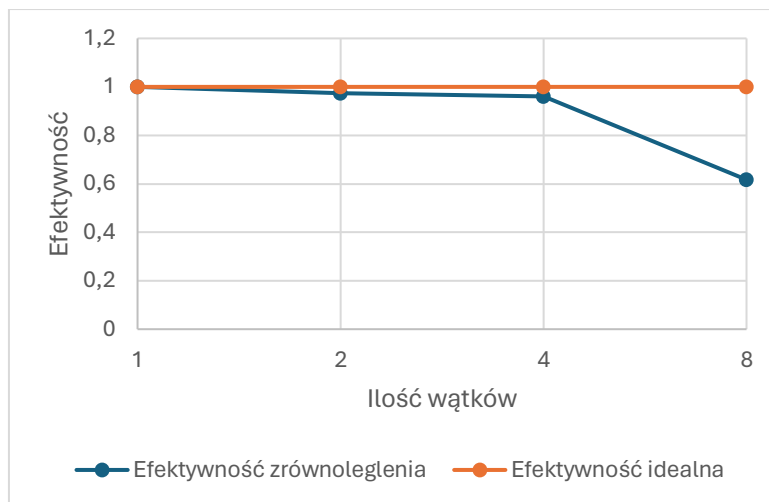
Czasu wykonania:



Przyspieszenia obliczeń:



Efektywności zrównoleglenia:



Następnie utworzono katalog `mat_vec`, gdzie rozpakowano pliki od prowadzącego i uruchomiono kod programu obliczania iloczynu macierz wektor w środowisku MPI dla wymiaru macierzy 12144:

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#include "mpi.h"
#include "pomiar_czasu.h"

#define WYMIAR 12144
#define ROZMIAR (WYMIAR*WYMIAR)

void mat_vec(double* a, double* x, double* y, int n, int nt);
```

Za pomocą parametru `-np` w pliku `Makefile` zmieniano liczbę wątków

```
# zależności i komendy
moj_program: moj_program.o mat_vec.o pomiar_czasu.o
$(LINKER) $(OPT) moj_program.o mat_vec.o pomiar_czasu.o -o moj_program $(LIB)
@echo 'To run execute "make run" (4 processes) or mplexec manually.'

moj_program.o: moj_program.c pomiar_czasu.h
$(CCOMP) -c $(OPT) moj_program.c $(INC)

mat_vec.o: mat_vec.c
$(CCOMP) -c $(OPT) mat_vec.c

pomiar_czasu.o: pomiar_czasu.c pomiar_czasu.h
$(CCOMP) -c $(OPT) pomiar_czasu.c

clean:
rm -f *.o moj_program

run: moj_program
$(MPI_run) -np 8 ./moj_program
```

Przykładowy wynik pomiaru dla 8 wątków:

```
loprych@loprych-VirtualBox:~/Desktop/PR_lab/lab_13/mat_vec/mat_vec_row_MPI_OpenM
l$ make
make: 'moj_program' is up to date.
loprych@loprych-VirtualBox:~/Desktop/PR_lab/lab_13/mat_vec/mat_vec_row_MPI_OpenM
l$ make run
mplexec -np 8 ./moj_program
Wersja rownolegla MPI/OpenMP
EXECUTION TIME: executing standard loop: 0.037641
Number of operations 294953472, PERFORMANCE 7.835953 GFlops
GBytes transferred to processor 1.179814, speed 31.343810 GB/s
```

Wszystkie wyniki zestawiono w pliku excel:

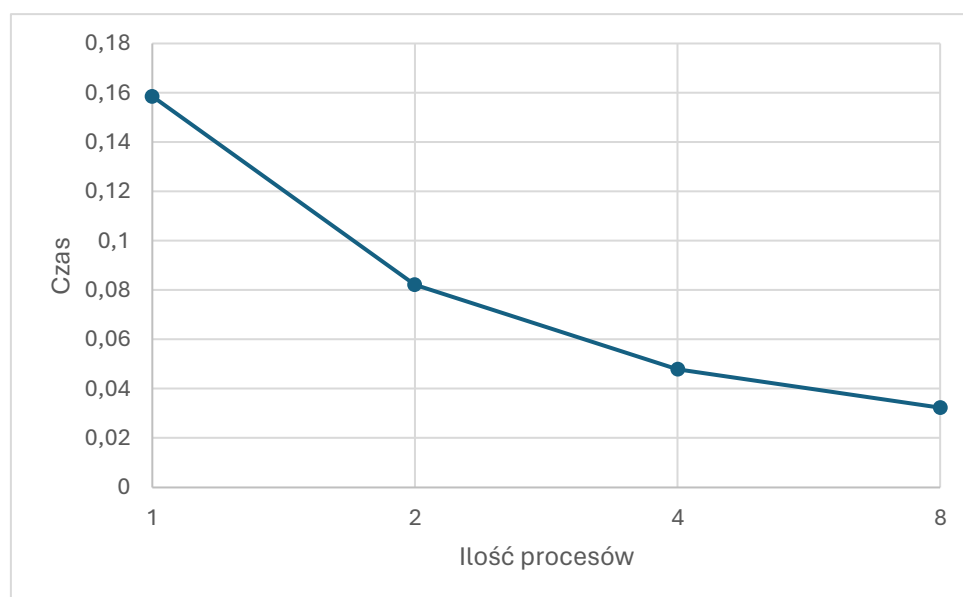
Nr pomiaru		l procesow = 1
1	0,15887	
2	0,15742	
3	0,15831	
Średni czas	0,15859	
Nr pomiaru		l procesow = 2
1	0,08174	
2	0,07962	
3	0,08267	
Średni czas	0,08221	
Nr pomiaru		l procesow = 4
1	0,0484	
2	0,04624	
3	0,04742	
Średni czas	0,04791	
Nr pomiaru		l procesow = 8
1	0,03384	
2	0,03405	
3	0,03086	
Średni czas	0,03235	

Następnie dokonano obliczeń dla efektywności i przyspieszenia względnego:

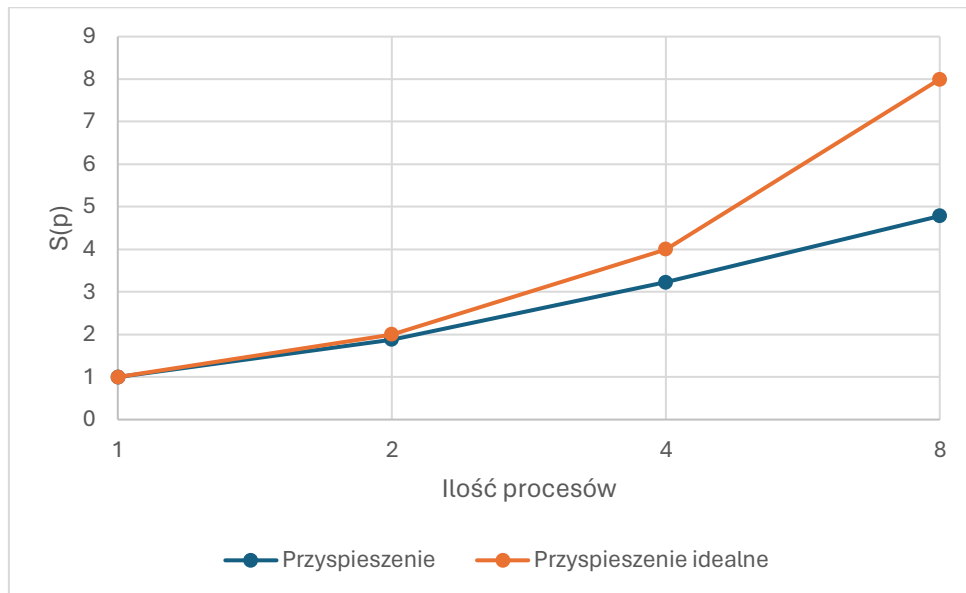
L. pocesow	Czas	Przyspieszenie	Efektynoc	Przyspiesz	Efektynosc idealna
1	0,15859	1	1	1	1
2	0,08221	1,88238693	0,94119	2	1
4	0,04791	3,229980066	0,8075	4	1
8	0,03235	4,783267905	0,59791	8	1

Następnie zestawiono wykresy zależności od liczby wątków dla:

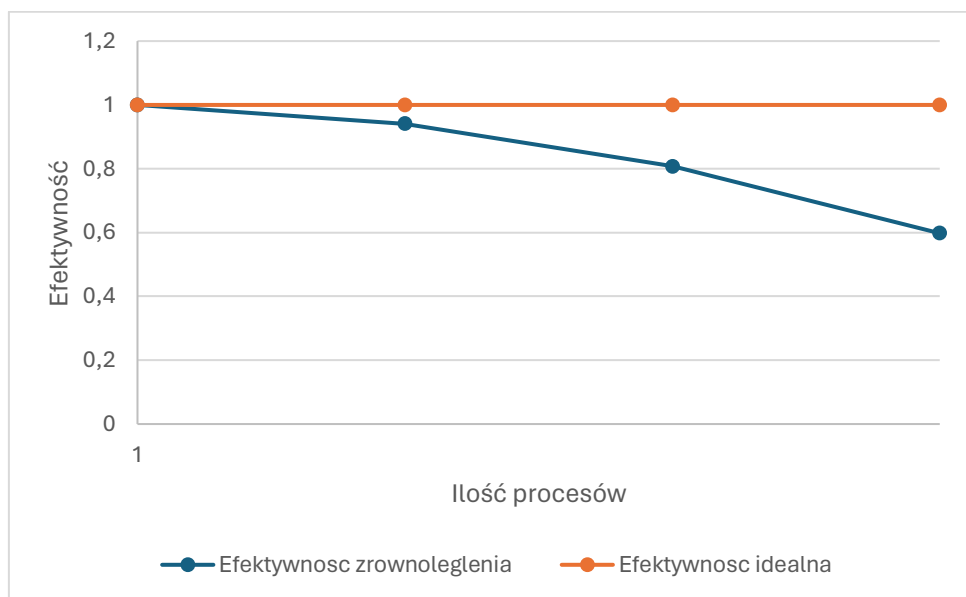
Czasu wykonania:



Przyspieszenia obliczeń:



Efektywności zrównoleglenia:



Analiza wyników

Czas wykonania

- zwiększając liczbę procesów/wątków obserwujemy spadek czasu wykonania
- najbardziej znaczący jest spadek przy przejściu z 1 na 2 procesy jak i wątki
- przeskok z 4 na 8 procesów jak i wątków daje bardzo niewielkie korzyści

Przyspieszenie

- cała:
 - zwiększając liczbę procesów/wątków rośnie przyspieszenie
 - dla dwóch procesów/wątków przyspieszenie jest bliskie idealnego
 - dalsze zwiększanie coraz bardziej oddala się od przyspieszenia idealnego
- wektor:
 - nie obserwujemy ciągłego wzrostu przyspieszenia
 - przy przeskoku z 4 na 8 procesów i wątków przyspieszenie maleje

Efektywność

- W obu przypadkach efektywność jest najwyższa dla 2 wątków oraz procesów
- Dalsze zwiększanie ilości wątków obniża efektywność

Podsumowanie

Rozważając dobór ilości wątków w OpenMP oraz procesów w przypadku MPI do wykonania naszego oprogramowania równoległego należy brać pod uwagę więcej niż jedną metrykę wydajności, bo nawet jeżeli spada czas to nie koniecznie musi to być robione w sposób wydajny.

Zbyt duża liczba procesów/wątków nie jest dobra, zasada im więcej tym lepiej tutaj nie działa. Dzieje się tak dlatego, że w pewnym momencie overhead tworzenia i zarządzania procesami /wątkami staje się większy niż zyski jakie dostajemy ze zrównoleglania.