Лабораторана робота 5

Створення та використання threadpool

Склад команди: Швець Анастасія, Фіалко Ярина
Репозиторій: https://github.com/ucu-cs/lab5_integral_threadpool-shvets_fialko_

Результати досліду:

```
$ python3 prog_runner.py 10 3 500
Function 1:
Result of integration: 4545450.0
Absolute error: 0.0230792
Relative error: 5.07743e-09
Min execution time: 137 ms
Avg execution time: 147.2 ms
Corrected sample standard deviation: 14.23 ms

Function 2:
Result of integration: 857208.0
Absolute error: 0.00117024
Relative error: 1.36517e-09
Min execution time: 10666 ms
Avg execution time: 11739.5 ms
Corrected sample standard deviation: 722.95 ms

Function 3:
Result of integration: -1.60467
Absolute error: 3.71871e-05
Relative error: 2.31743e-05
Min execution time: 158 ms
Avg execution time: 193.7 ms
Corrected sample standard deviation: 30.60 ms
```

THREADPOOL

	function 1	function 2	function 3
Min execution time	137 ms	10666 ms	158 ms
Avg execution time	147.2 ms	11739.5 ms	193.7 ms
Corrected sample standard deviation	14.23 ms	722.95 ms	30.60 ms

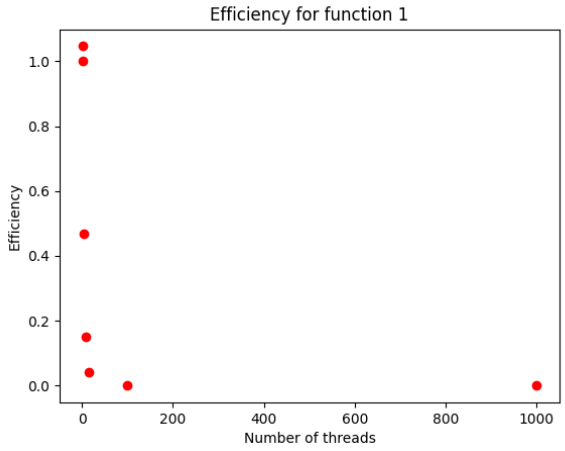
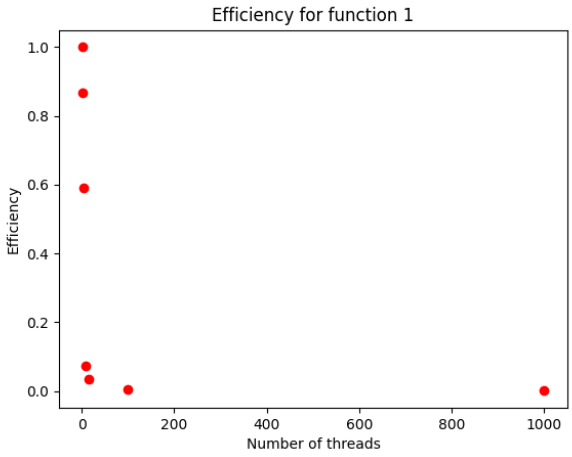
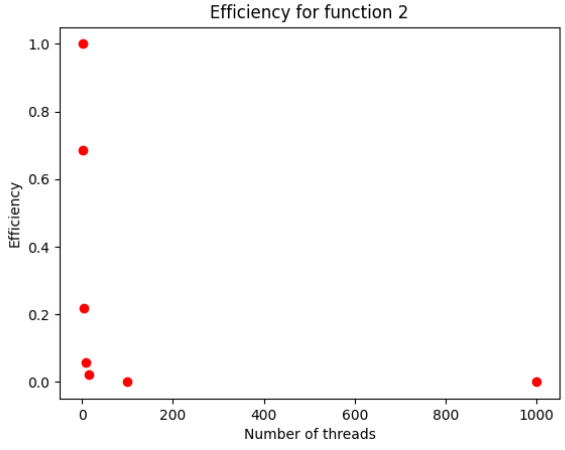
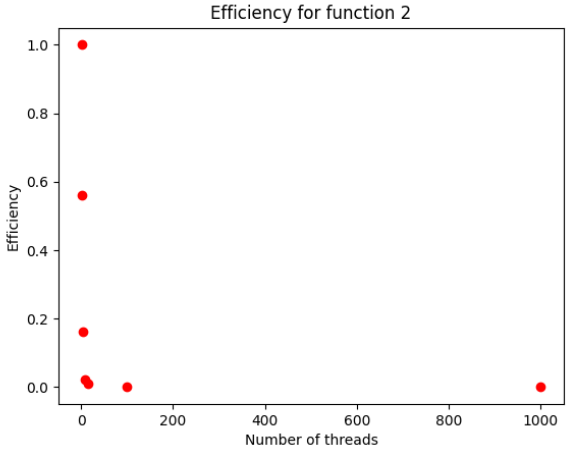
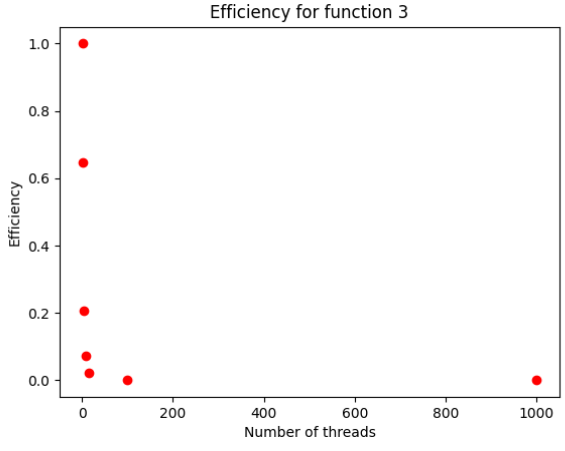
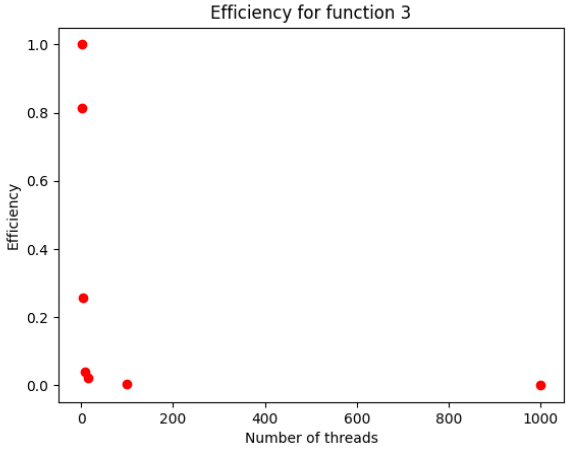
THREAD-SAFE QUEUE

	function 1	function 2	function 3
Min execution time	101 ms	4104 ms	78 ms
Avg execution time	109.0 ms	4205.0 ms	89.0 ms
Corrected sample standard deviation	9.17 ms	87.48 ms	9.64 ms

FUCNTION 1		
threads	threadpool	thread-safe queue
1	419 ms	116 ms
2	193 ms	86 ms
4	240 ms	59 ms
8	411 ms	338 ms
16	550 ms	327 ms
100	2401 ms	304 ms
1000	18083 ms	385 ms

FUCNTION 2		
threads	threadpool	thread-safe queue
1	14213 ms	1256 ms
2	9692 ms	1366 ms
4	15685 ms	1798 ms
8	29110 ms	20325 ms
16	38872 ms	20430 ms
100	111730 ms	19055 ms
1000	1605204 ms	18035

FUCNTION 3		
threads	threadpool	thread-safe queue
1	265 ms	32 ms
2	168 ms	29 ms
4	268 ms	37 ms
8	394 ms	328 ms
16	515 ms	408 ms
100	2990 ms	321 ms
1000	8074 ms	

threadpool	thread-safe queue
	
	
	

Аналіз/Висновок:

Із зображених вище порівняльних таблиць та графіків, можна зробити висновок про значну перевагу у використанні потокобезпечної черги проти threadpool у задачі інтегрування. Попри те, що при використанні threadpool вдалось оптимізувати робот у потоків, уникнувши їх постійному видалянню та створенню, цей варіант реалізації паралелізму видався менш вдалим через необхідність використання std::future для отримання результату обрахунку функції у кожній точці розбиття. Припускаємо, що threadpool покаще свою ефективність для задач які не вимагають повернення результату атомарної задачі.