

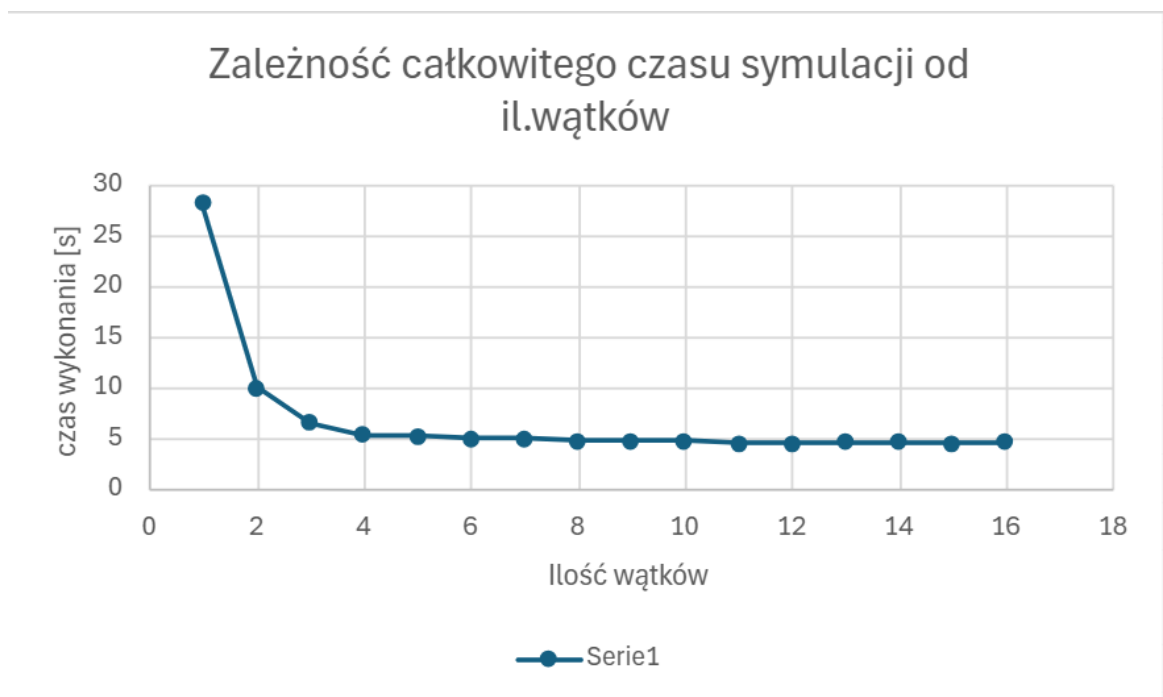
Analiza efektywności wielowątkowości w aplikacji symulującej rozprzestrzenianie infekcji

1. Wykresy czasów wykonania w zależności od liczby wątków

Do analizy efektywności wykonano testy (500000 danych) dla liczby wątków od 1 do 16. Zmierzono trzy parametry:

- **czas całkowity wykonania programu,**
- **średni czas fazy 1** (symulacja infekcji),
- **średni czas fazy 2** (aktualizacja danych).

Na podstawie danych przygotowano trzy wykresy:





•



•

Widzimy wyraźnie, że czas wykonania skraca się do około 8 wątków, a następnie stabilizuje się.

2. Opis zakresu użycia wielowątkowości i logiki przetwarzania danych

Program dzieli dane populacji na równe fragmenty, z którymi pracują osobne wątki. Każdy wątek realizuje:

- **faza 1:** symulacja zakażeń,

- **faza 2:** aktualizacja statusu chorych i ozdrowień.

Użyto synchronizacji (`Lock`, `Condition`) do sterowania przejściami między fazami, aby zapewnić spójność działania i zapobiec wyścigom wątków.

Program operuje w klasie `Simulation`, gdzie wątki działają na przydzielonych zakresach indeksów. Zakażeni są tymczasowo przechowywani lokalnie w `infectedList`, co zapobiega konfliktom w dostępie do danych.

Synchronizacja pomiędzy fazami pozwala wszystkim wątkom zakończyć daną fazę przed rozpoczęciem kolejnej, zapewniając spójność i deterministyczny przepływ symulacji.

3. Wnioski z analizy uzyskanych wyników

Największe zyski czasowe zauważono do 8 wątków, co pokrywa się z liczbą logicznych rdzeni procesora (Intel Core i5-1135G7). Powyżej tej liczby efektywność spada, a przyrosty są marginalne. Faza 1 daje największy zysk z równoległości, natomiast faza 2 pozostaje praktycznie niezmienna i odporna na zmianę liczby wątków (wynika to z tego, że uruchamiając fazę 2, wywołujemy funkcję `updatePerson`, w której używamy `synchronized`). To ogranicza możliwość jednoczesnej modyfikacji danych przez wiele wątków, co wpływa na brak skalowalności tej fazy.

Przy większej liczbie wątków widać też drobne fluktuacje wyników, co potwierdza wpływ zarządzania zasobami przez system operacyjny.

4. Ograniczenia i zakres stosowalności wielowątkowości

Wielowątkowość jest korzystna w przypadku dużych danych i liczby wątków zbliżone do liczby rdzeni. W przeciwnym razie pojawiają się:

- narzuty synchronizacyjne,
- ograniczenia pamięci podręcznej,
- konflikty przy współdzielonych danych.

Dlatego przed zastosowaniem wielowątkowości należy przeprowadzić analizę, jak ta powyższa, by dobrać optymalną liczbę wątków i skalę danych.