Teoria współbieżności

Laboratorium 3
Problem Pięciu Filozofów – różne rozwiązania

Natalia Bratek
14.11.2024

Spis treści

1. Polecenie

Zaprojektuj algorytm jednoczesnej alokacji współdzielonych zasobów (widelce) przez konkurujące procesy (filozofowie), tak aby uniknąć zakleszczenia i zagłodzenia.

2. Rozwiązanie naiwne (z możliwością blokady)

2.1 Opis implementacji

W klasie NaivePhilosopher każdy filozof najpierw wykonuje think(), a następnie próbuje uzyskać dostęp do widelców przez synchronizację. Gdy filozof jest gotowy do jedzenia, próbuje uzyskać dostęp do lewego widelca za pomocą synchronized. Następnie próbuje podnieść prawy widelec w kolejnym synchronized. Jeśli oba widelce są dostępne, filozof oblicza czas oczekiwania na widelce i przechodzi do jedzenia. Po zakończeniu jedzenia odkłada prawy widelec, a następnie lewy.

Rozwiązanie naiwne jest podatne na zakleszczenie.

2.2 Wyniki

Przykład dla 5 filozofów. Potencjalny problem dla synchronizacji, zakleszczania.

```
Filozof 4 podniósł lewy widelec.
Filozof 3 podniósł lewy widelec.
Filozof 2 podniósł lewy widelec.
Filozof 1 podniósł lewy widelec.
```

Rys. 1 Problem dla rozwiązania naiwnego

Każdy z filozofów (Filozofowie 4, 3, 2, 1) podniósł swój lewy widelec. Jeśli Filozof 5 również podniesie swój lewy widelec, to wszyscy filozofowie będą czekać na prawy widelec, który jest aktualnie zajęty przez sąsiada. Oznacza to, że żaden z filozofów nie może

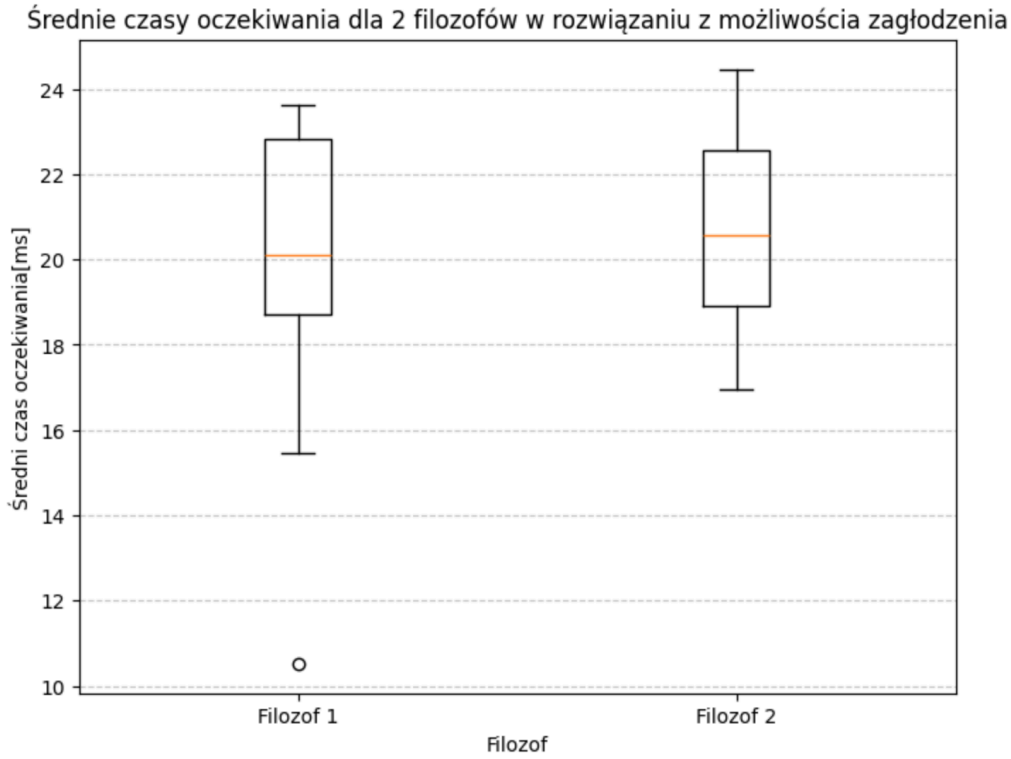
kontynuować, ponieważ każdy z nich czeka na zasób, który nigdy nie zostanie zwolniony.

3. Rozwiązanie z możliwością zagłodzenia

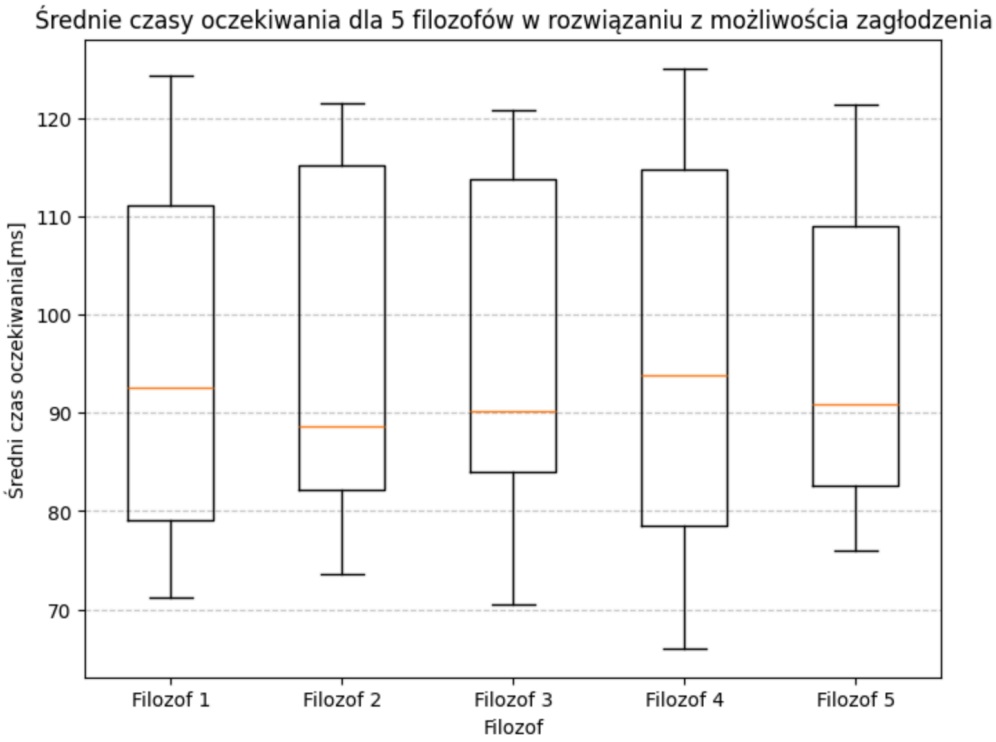
3.1 Opis implementacji

Klasa StarvationPhilosopher przedstawia filozofa, który kiedy chce jeść, najpierw próbuje uzyskać dostęp do lewego widelca za pomocą synchronized, co wskazuje, że lewy widelec jest dostępny. Następnie próbuje zdobyć prawy widelec, również za pomocą synchronized. Jeśli uda mu się zdobyć oba widelce, filozof oblicza czas oczekiwania i przechodzi do jedzenia. Jeśli nie uda się zdobyć obu widelców (czyli synchronized się nie powiedzie), filozof czeka przez losowy czas (Thread.sleep), a następnie ponawia próbę.

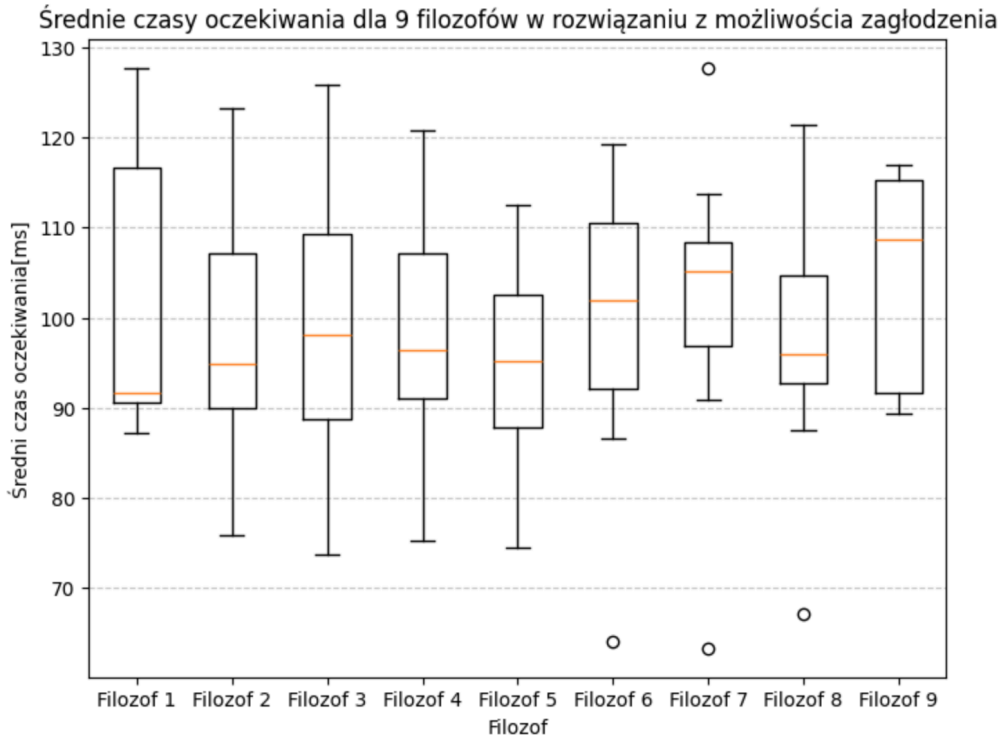
3.2 Wykresy



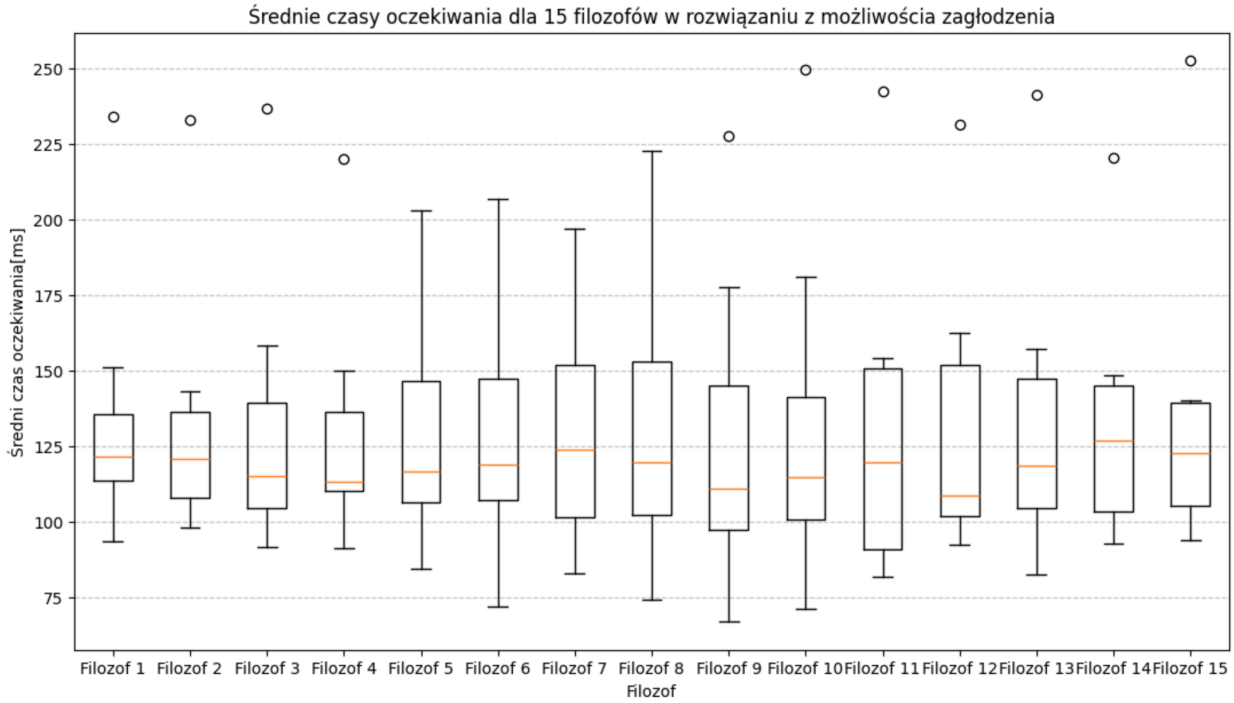
Rys.2 Wykres pudełkowy dla 2 filozofów



Rys.3 Wykres pudełkowy dla 5 filozofów



Rys.4 Wykres pudełkowy dla 9 filozofów



Rys.5 Wykres pudełkowy dla 15 filozofów

3.3 Wnioski

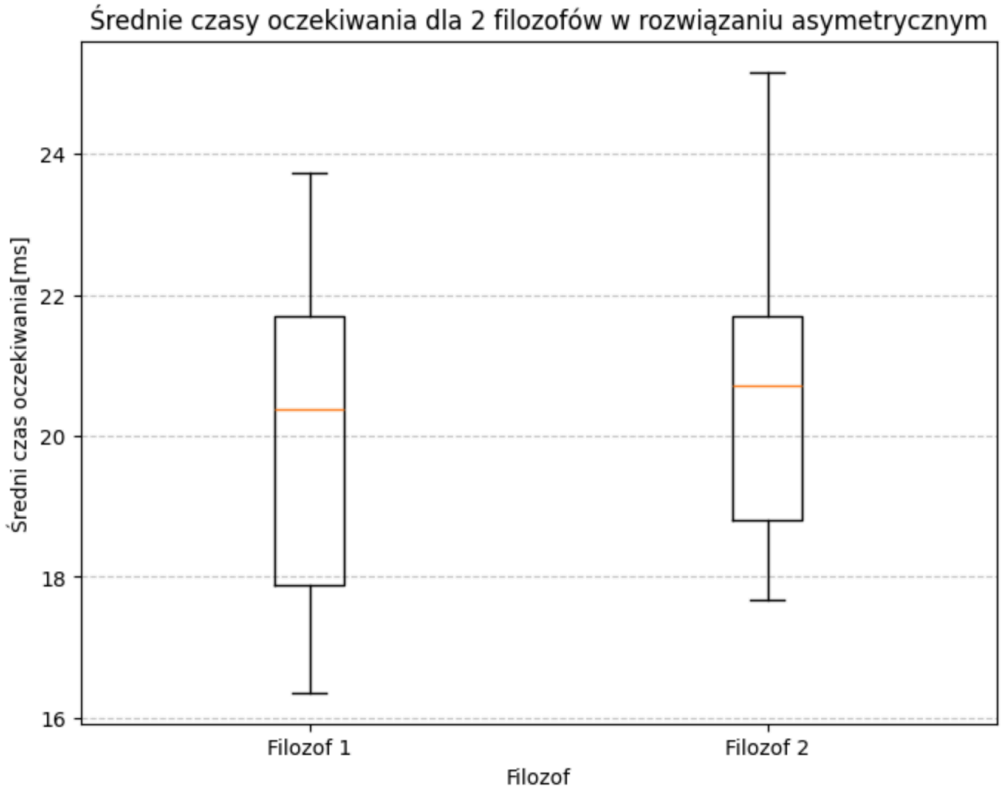
- Mediana czasów oczekiwania dla dwóch filozofów jest stosunkowo niska i dość wyrównana, co może wskazywać na bardziej spójny dostęp do zasobów.
- Dla większej liczby filozofów, jak 9 czy 15, widać, że czasem niektórzy filozofowie musieli czekać trochę dłużej (wartości odstające), co może być spowodowane tym, że przez pewien czas nie mieli dostępu do widelców.
- Wraz ze wzrostem liczby filozofów rośnie zakres czasów oczekiwania, co pokazuje większą konkurencję o zasoby.
- Pomimo że to rozwiązanie może prowadzić do zagłodzenia filozofów, wyniki pokazały, że w tej symulacji wszyscy filozofowie mieli wystarczająco częsty dostęp do widelców, co oznacza, że zagłodzenie nie wystąpiło – pewnie dlatego, że symulacja była za krótka (trwała 10 sekund).

4. Rozwiązanie asymetryczne

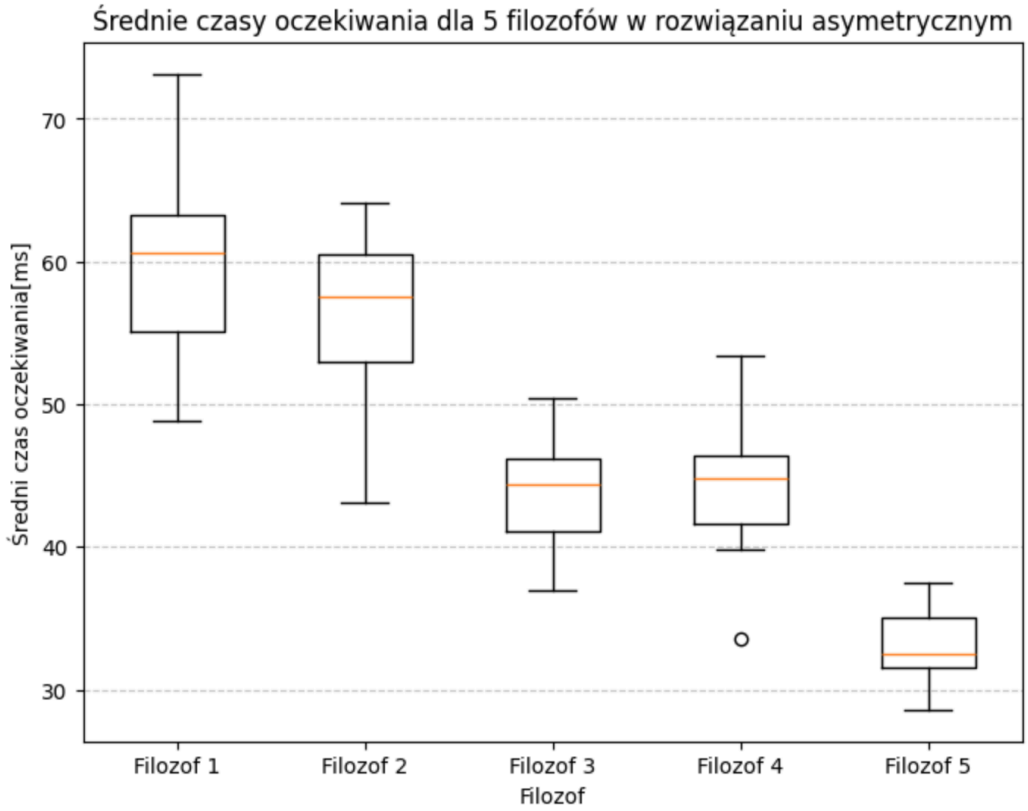
4.1 Opis implementacji

Filozofowie, którzy mają parzysty numer (indeks) zamieniają kolejność podnoszenia widelców – najpierw próbują podnieść prawy widelec, a następnie lewy. Filozofowie o nieparzystych numerach (indeksach) podnoszą widelce w kolejności: najpierw lewy, a potem prawy.
Filozof rozpoczyna od myślenia, po której przechodzi do próby zdobycia obu widelców, korzystając z synchronized. Czas oczekiwania na zdobycie widelców jest obliczony. Po zdobyciu obu widelców filozof przystępuje do jedzenia, a następnie odkłada widelce, zwalniając dostęp dla innych filozofów.

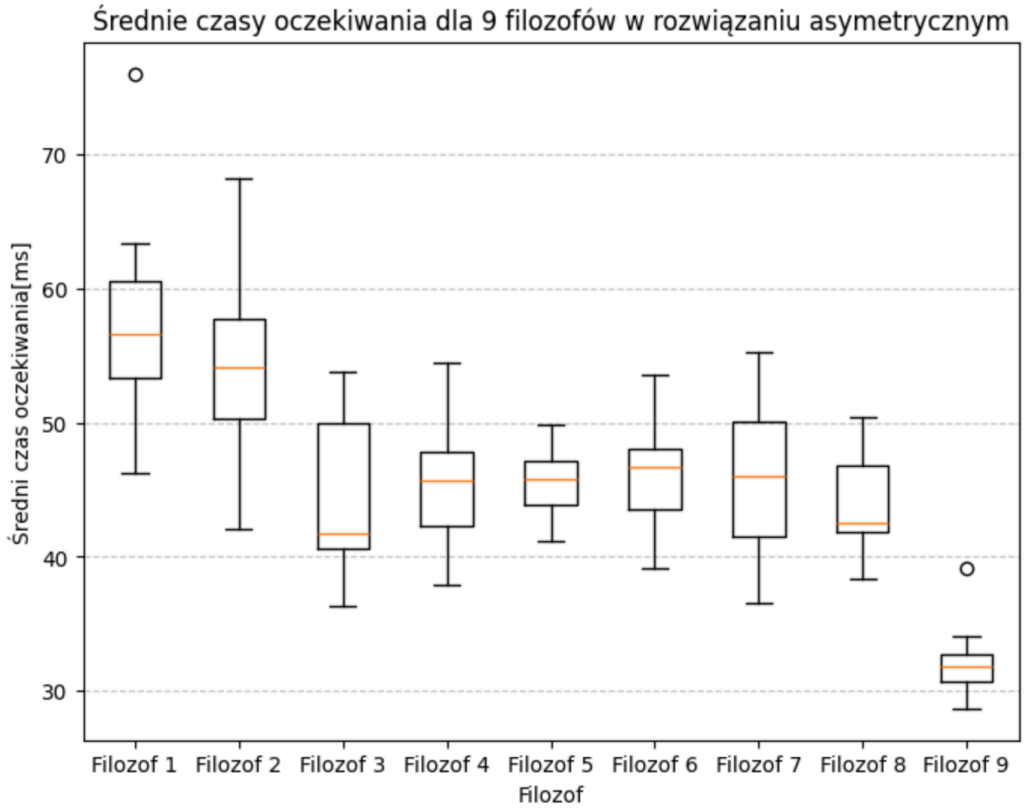
4.2 Wykresy



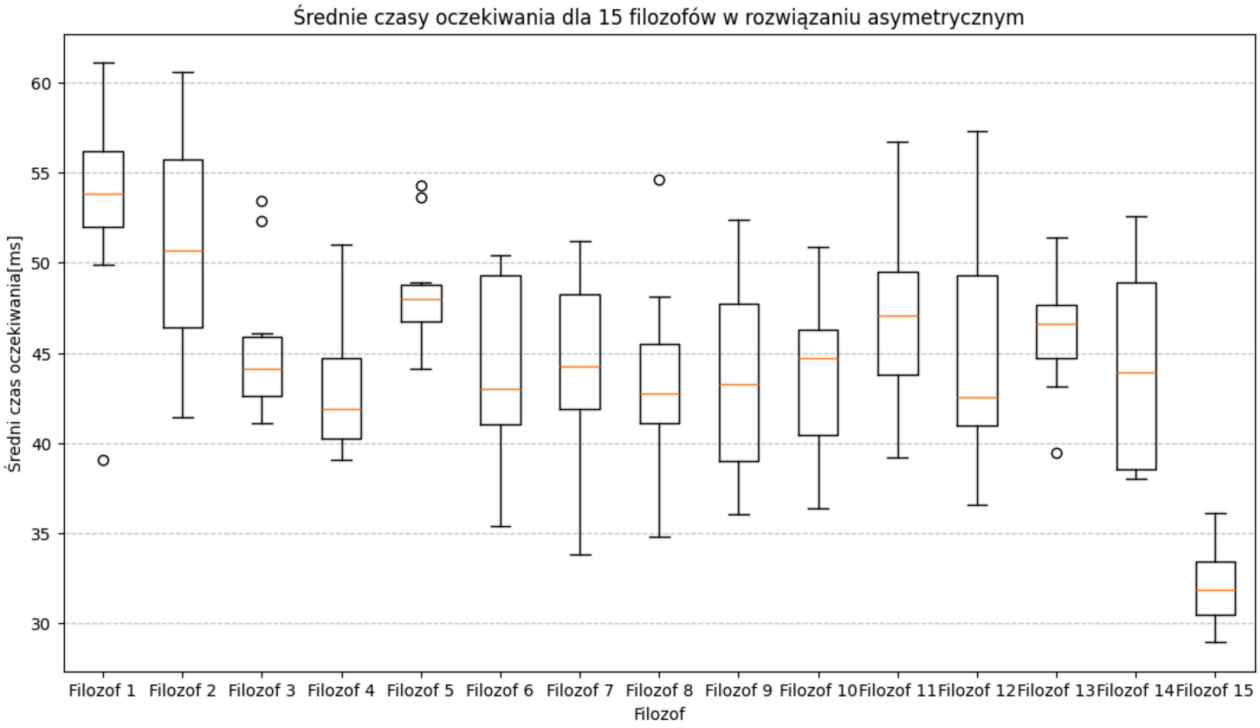
Rys.6 Wykres pudełkowy dla 2 filozofów



Rys.7 Wykres pudełkowy dla 5 filozofów



Rys.8 Wykres pudełkowy dla 9 filozofów



Rys.9 Wykres pudełkowy dla 15 filozofów

4.3 Wnioski

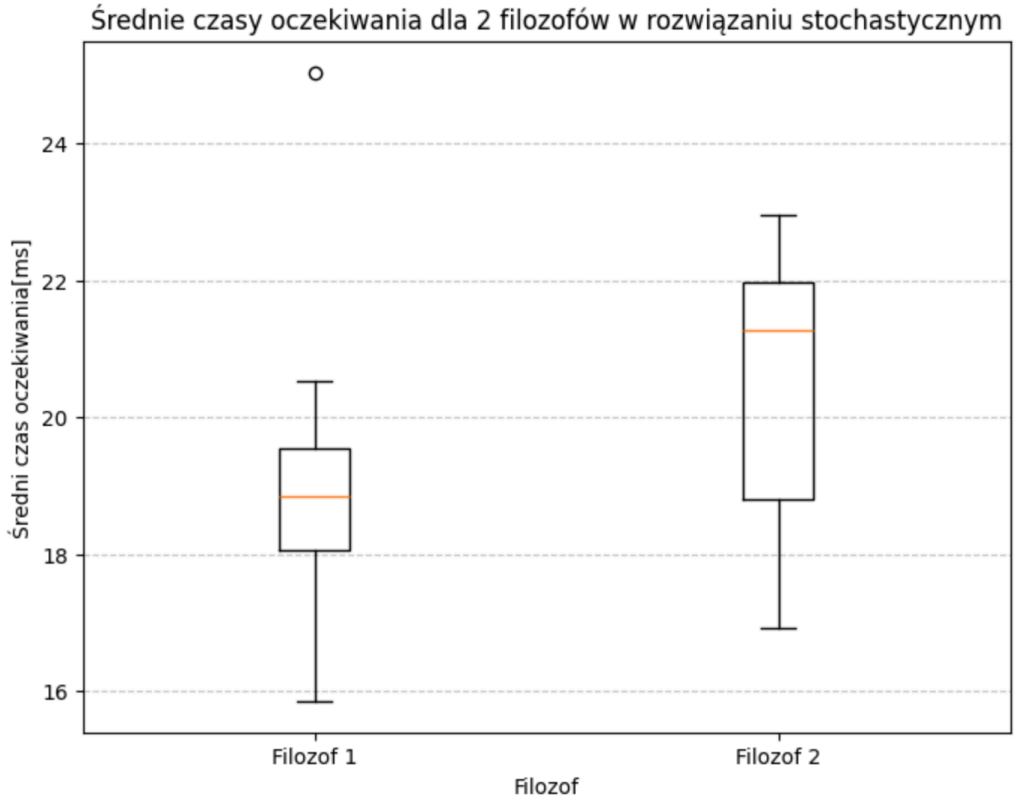
- Na wykresach widoczne jest, że nie wszyscy filozofowie oczekują tyle samo.
- W rozwiązaniach dla większej liczby filozofów można zauważyć przypadki ekstremalnie długiego czasu oczekiwania (wartości odstające).
- Wysoka rozpiętość między wartościami minimalnymi a maksymalnymi wskazuje na dużą rozbieżność w dostępie do zasobów.

5. Rozwiązanie stochastyczne

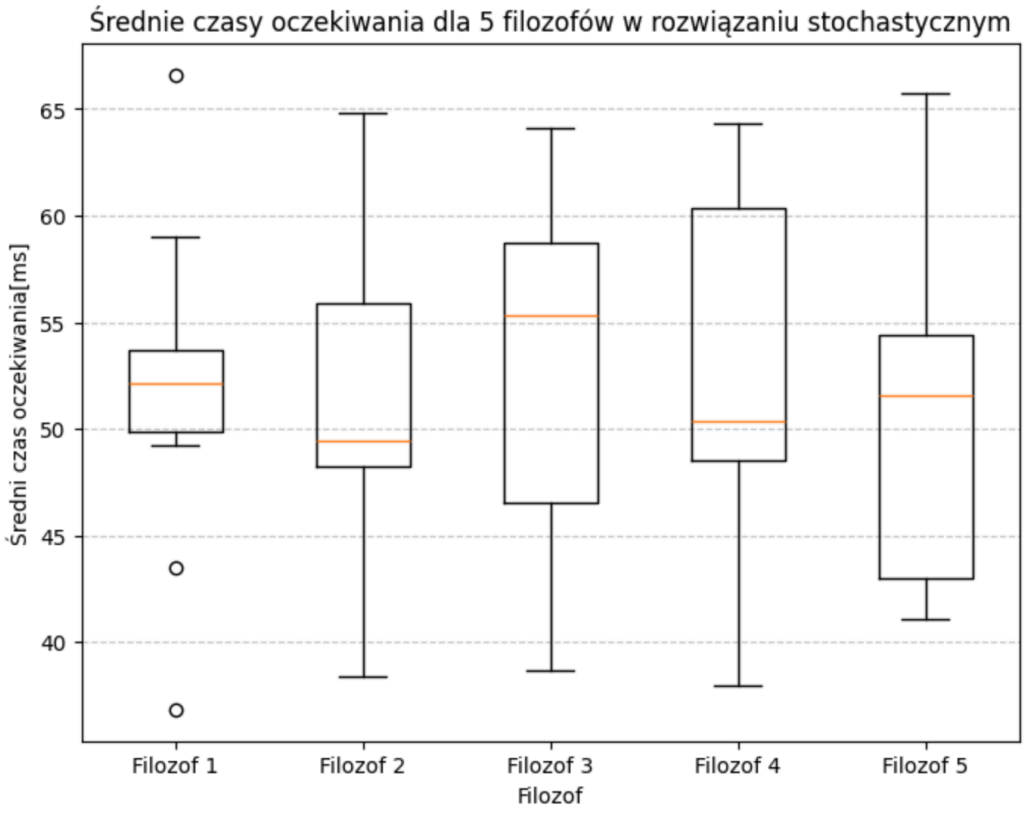
5.1 Opis implementacji

W implementacji rozwiązania stochastycznego każdy filozof losowo decyduje, od którego widelca zacząć. Za pomocą obiektu Random ustalana jest kolejność – filozof albo próbuje najpierw podnieść lewy widelec, a potem prawy, albo odwrotnie. Filozofowie próbują zdobyć widelce za pomocą synchronized. Czas oczekiwania na zdobycie obu widelców jest obliczany. Po zdobyciu obu widelców filozof przechodzi do jedzenia, a po skończonym posiłku odkłada widelce, zwalniając je dla innych filozofów.

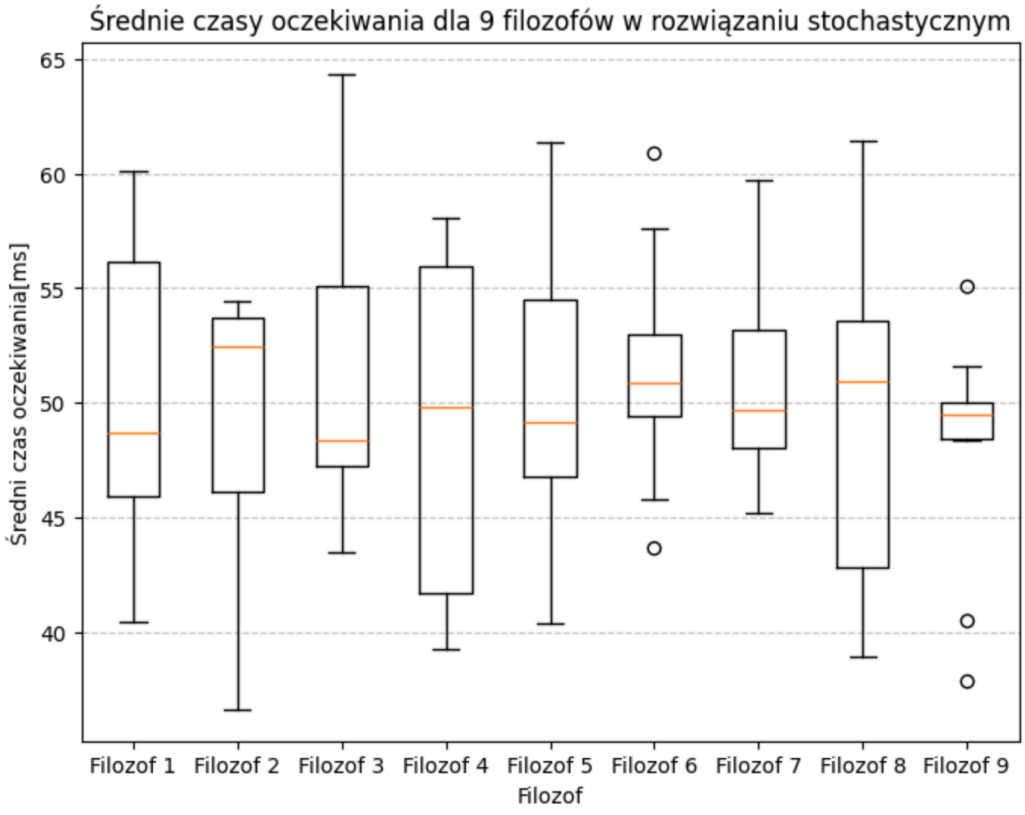
5.2 Wykresy



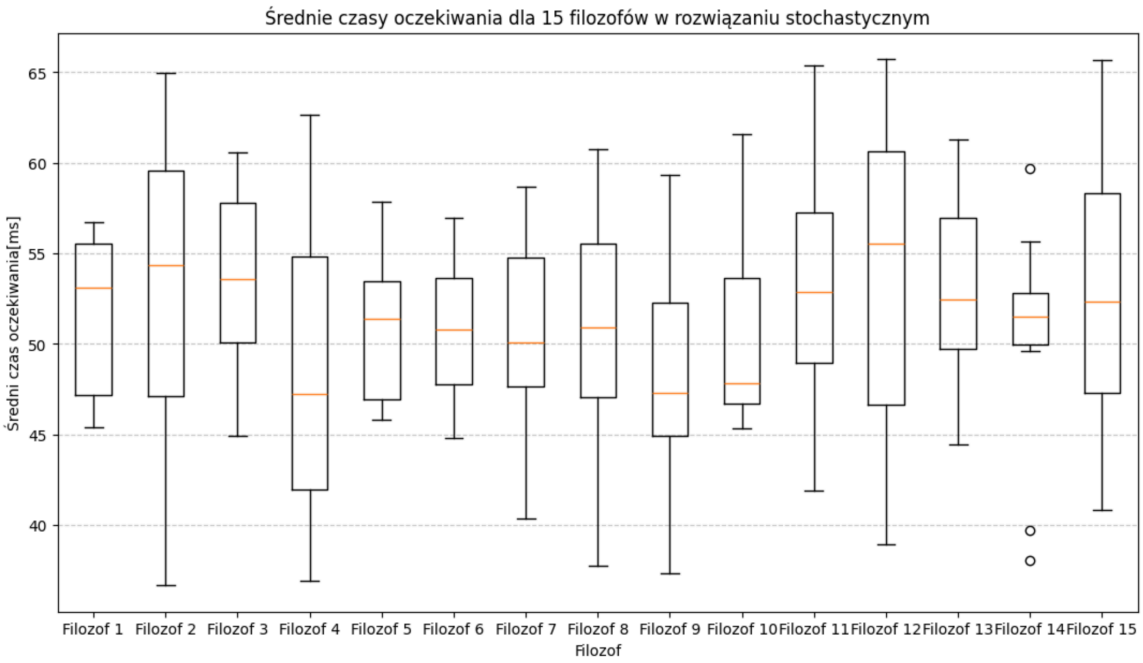
Rys.10 Wykres pudełkowy dla 2 filozofów



Rys.11 Wykres pudełkowy dla 5 filozofów



Rys.12 Wykres pudełkowy dla 9 filozofów



Rys.13 Wykres pudełkowy dla 15 filozofów

5.3 Wnioski

- W niektórych wykresach można zaobserwować wartości odstające, co oznacza, że czasami pojedynczy filozof czekał znacznie dłużej niż pozostali.
- Wraz ze wzrostem liczby filozofów, zakresy czasów oczekiwania ulegają znacznemu zwiększeniu.
- Losowe wybieranie kolejności podnoszenia widelców prowadzi do większej zmienności w czasie oczekiwania

6. Rozwiązanie z arbitrem

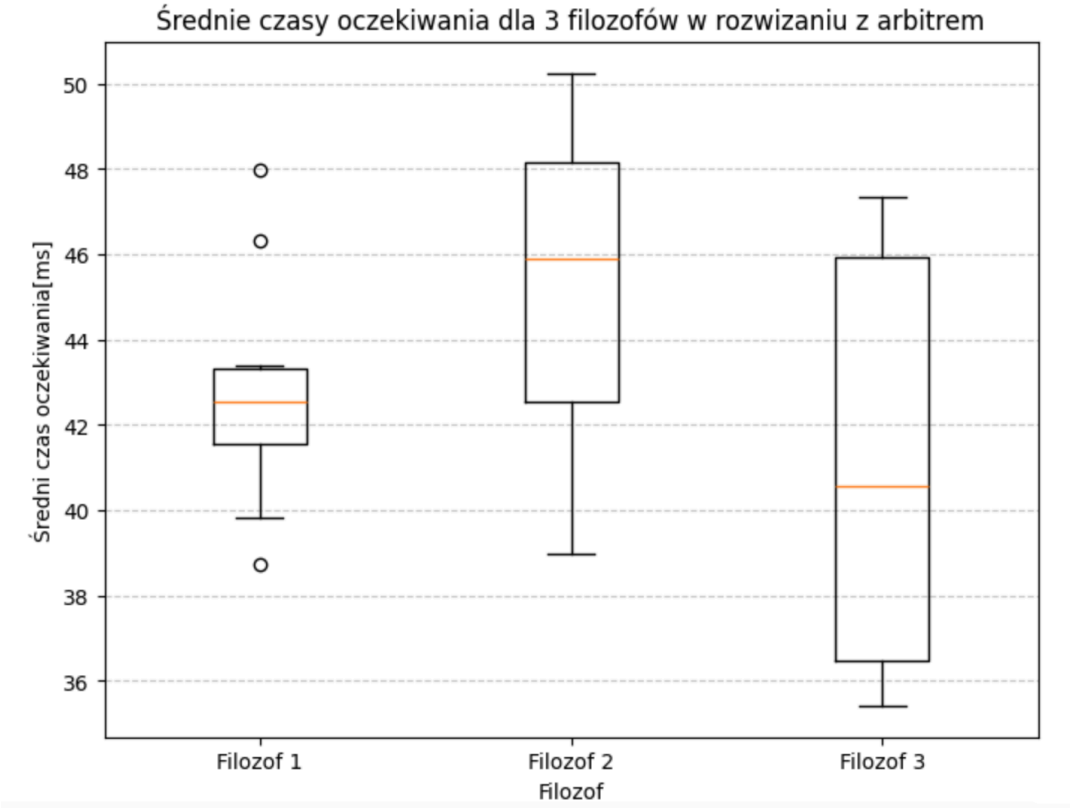
6.1 Opis implementacji

W implementacji z arbitrem, każdy filozof przed próbą zdobycia widelców musi najpierw uzyskać pozwolenie od lokaja, reprezentowanego przez semafor. Semafor ogranicza liczbę filozofów, którzy mogą jednocześnie próbować jeść. Gdy filozof uzyska dostęp (acquire) od lokaja, oblicza czas oczekiwania i próbuje zablokować oba widelce za pomocą synchronized. Po zakończeniu jedzenia filozof zwalnia dostęp do widelców i oddaje pozwolenie (release) lokajowi, umożliwiając kolejnym filozofom dostęp do stołu.

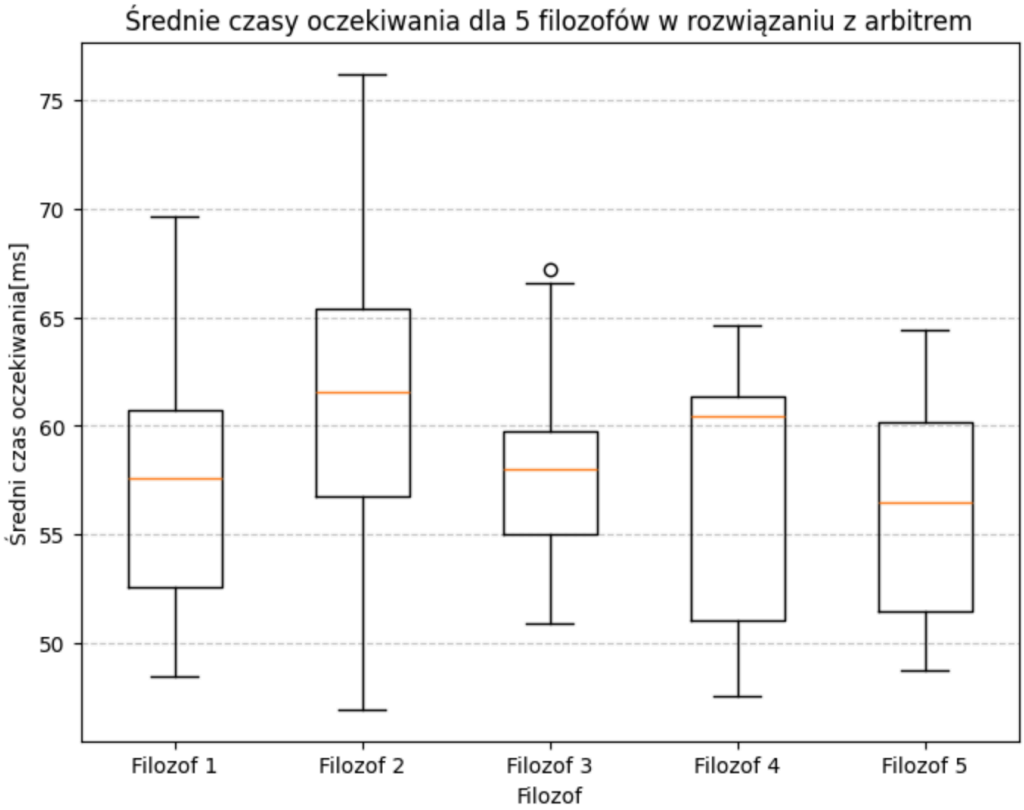
Takie rozwiązanie zmniejsza ryzyko zakleszczania.

6.2 Wykresy

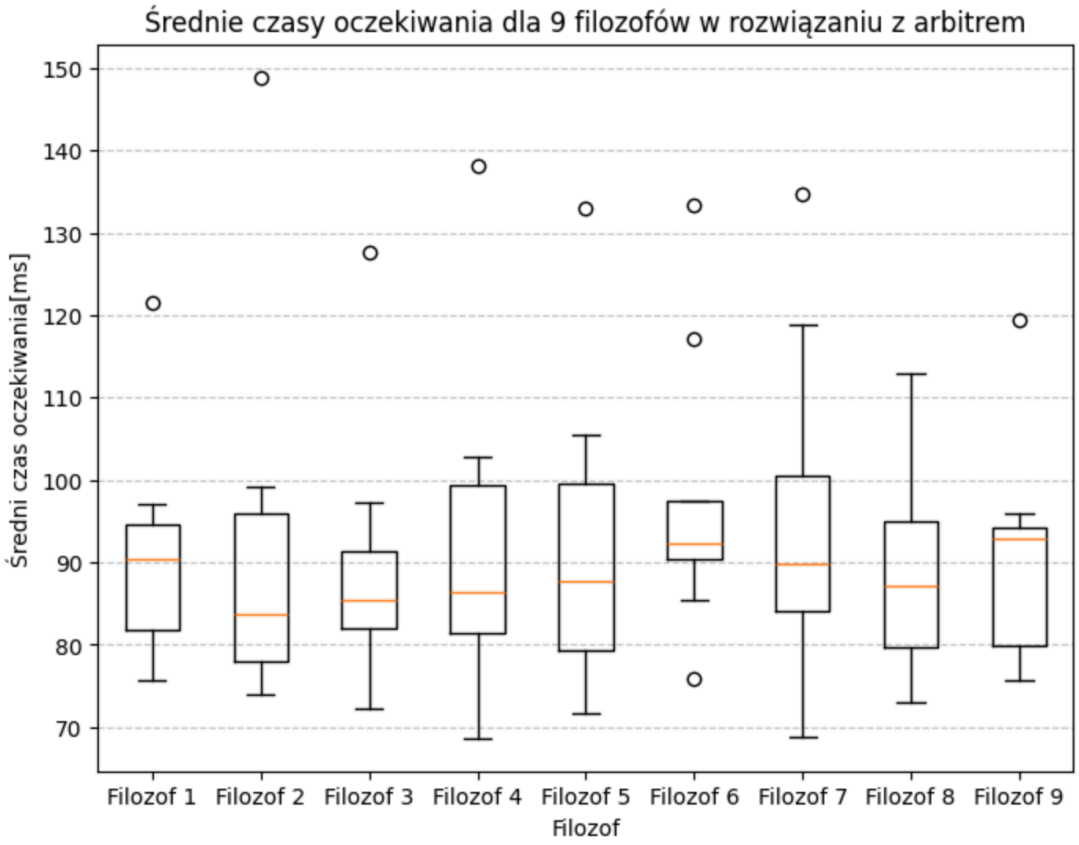
Z powodu szybkiego uzyskiwania dostępu – dwóch filozofów uzyskało dostęp do widelców bardzo szybko (np. z powodu małej liczby filozofów lub niskiej konkurencji), co spowodowało, że czas oczekiwania był równy lub bliski zeru. Dlatego zdecydowałam się wykonać pomiary i wykresy dla trzech filozofów.



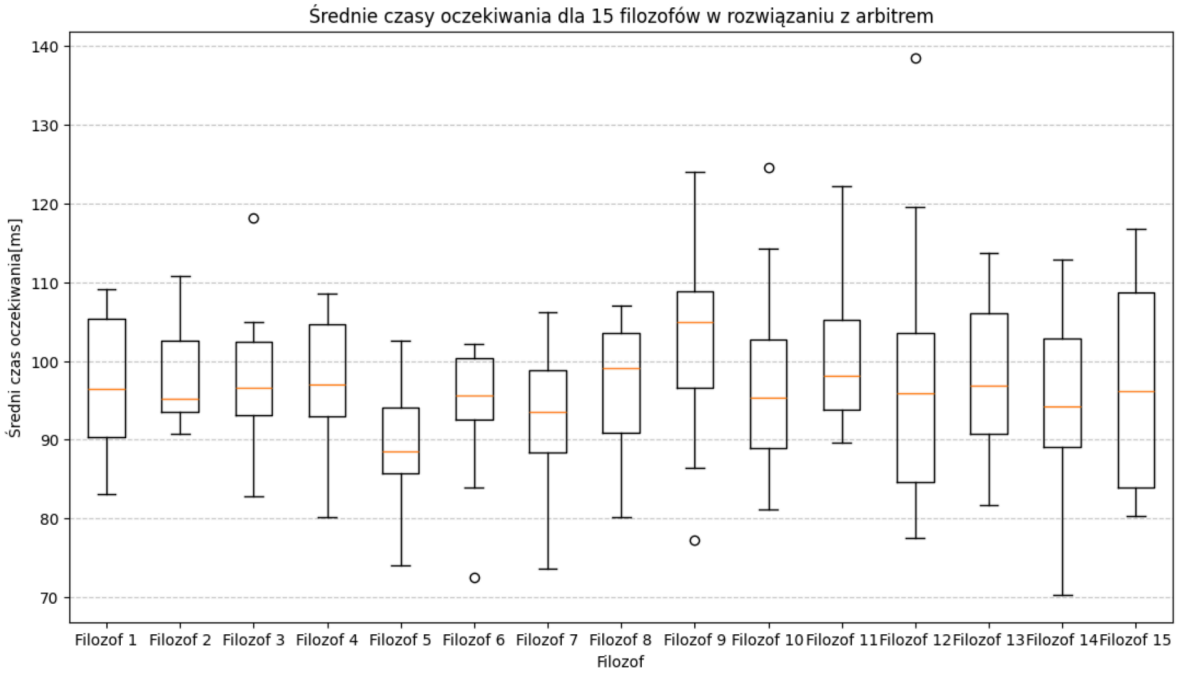
Rys.14 Wykres pudełkowy dla 3 filozofów



Rys.15 Wykres pudełkowy dla 5 filozofów



Rys.16 Wykres pudełkowy dla 9 filozofów



Rys.17 Wykres pudełkowy dla 15 filozofów

6.3 Wnioski

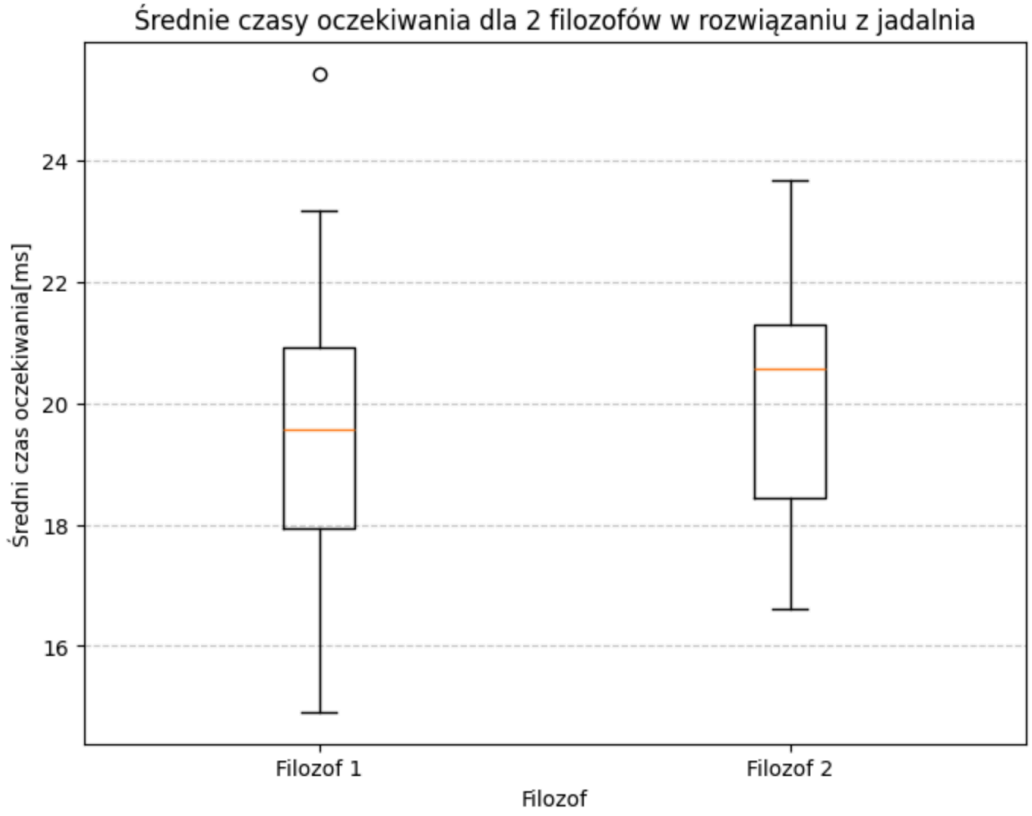
- Wraz ze wzrostem liczby filozofów średnie czasy oczekiwania na dostęp do zasobów ulegają zwiększeniu.
- Arbiter, ograniczając liczbę jednocześnie próbujących jeść filozofów, zmniejsza ryzyko zakleszczenia.
- Wraz ze wzrostem liczby filozofów można zauważyć bardziej stabilne rozkłady czasu oczekiwania. Wskazuje to na to, że arbiter jest skuteczny w zmniejszaniu skrajnych przypadków czekania.
- Większość filozofów osiąga relatywnie podobne czasy, co sugeruje na stabilność

7. Rozwiązanie z jadalnią

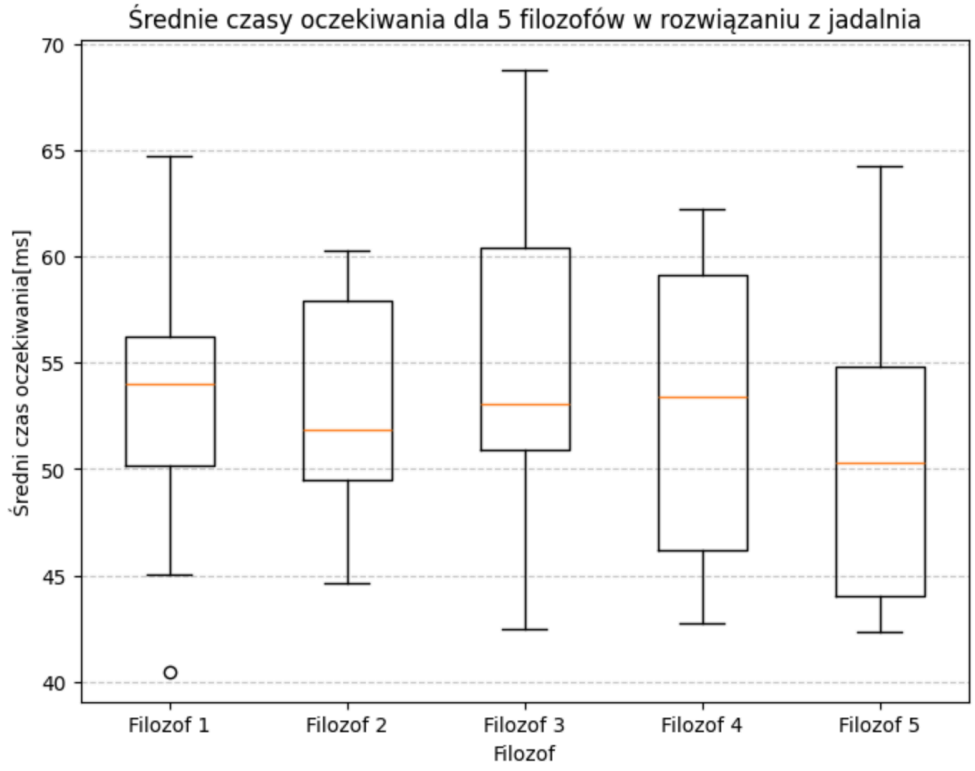
7.1 Opis implementacji

Filozofowie, zanim przystąpią do zdobycia widelców, sprawdzają możliwość wejścia do jadalni, korzystając z semafora (lokaj.tryAcquire()). Jeśli uzyskają dostęp do jadalni, próbują zdobyć lewy widelec, a następnie prawy widelec za pomocą synchronized i jedzą, obliczając czas oczekiwania. Po zakończeniu posiłku zwalniają semafor (lokaj.release()). W przypadku braku dostępu do jadalni filozof próbuje zdobyć widelce w odwrotnej kolejności (najpierw prawy, a potem lewy).

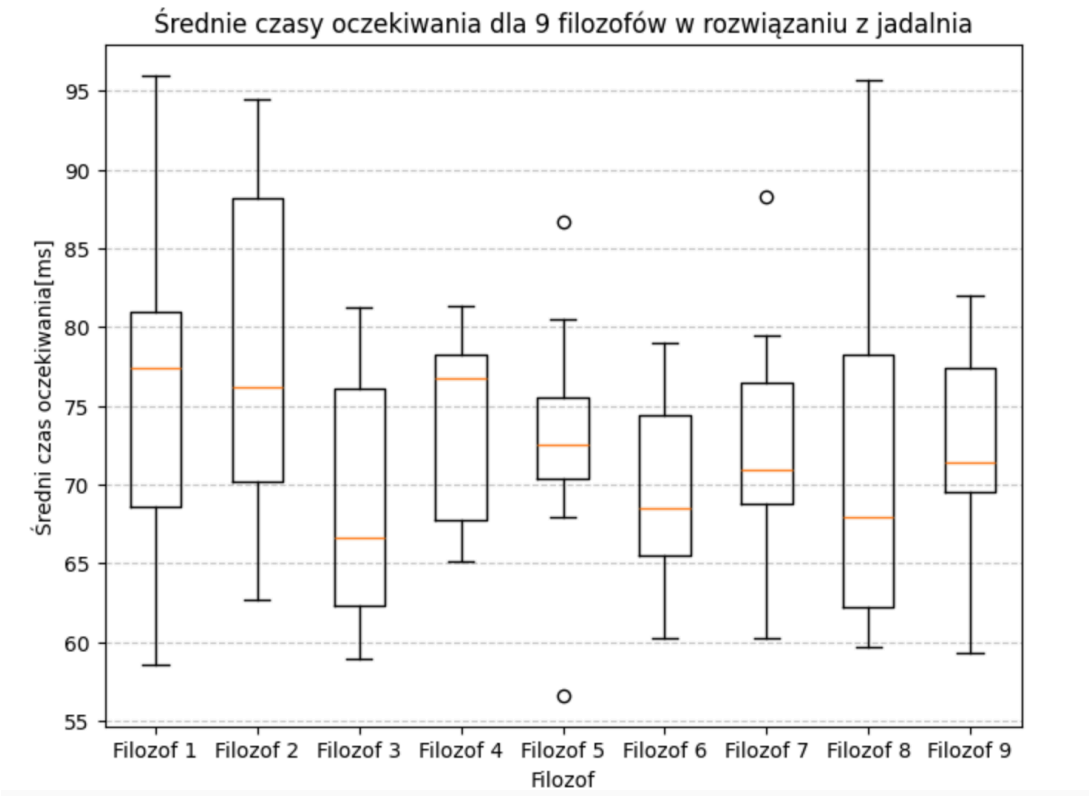
7.2 Wykresy



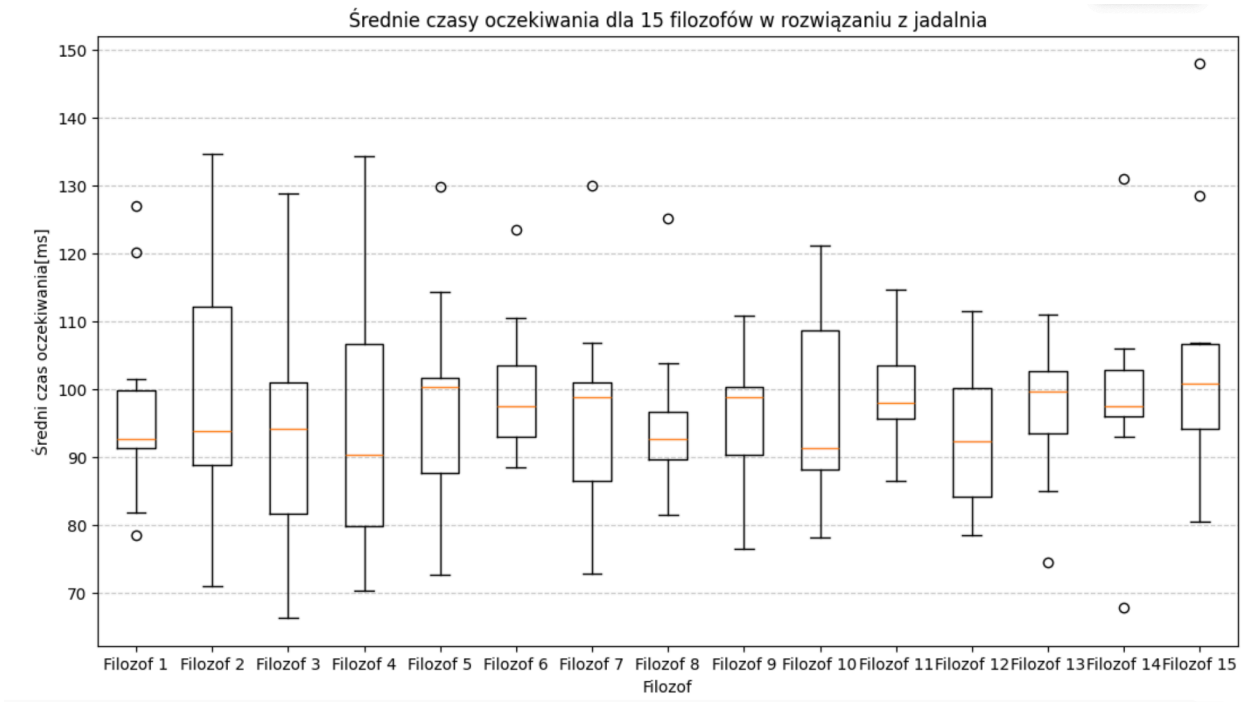
Rys.18 Wykres pudełkowy dla 2 filozofów



Rys.18 Wykres pudełkowy dla 5 filozofów



Rys.19 Wykres pudełkowy dla 9 filozofów



Rys.20 Wykres pudełkowy dla 15 filozofów

7.3 Wnioski

- Dla małej liczby filozofów średnie czasy oczekiwania są zbliżone i relatywnie niskie
- W miarę zwiększania liczby filozofów można zaobserwować rosnącą konkurencję.
- Pomimo rosnącej liczby filozofów, mediana czasu oczekiwania pozostaje w miarę stabilna.
- Rozwiązanie z jadalnią zmniejsza ryzyko zakleszczenia, ale nie eliminuje go całkowicie.

8. Ogólne wnioski

- Rozwiązanie z możliwością zagłodzenia nie jest optymalne dla dużej liczby filozofów, ponieważ istnieje ryzyko, że niektórzy z nich będą znacznie dłużej czekać na dostęp do zasobów (zagłodzenie)
- Rozwiązania naiwne ma największe ryzyko zakleszczenia np. gdy wszyscy filozofowie próbują jednocześnie podnieść widelce, blokując się wzajemnie
- Obecność arbitra zapewnia, że nie dochodzi do zakleszczenia, ale czasem może prowadzić do sytuacji, w której jeden filozof czeka nieco dłużej z powodu ograniczeń nałożonych przez semafor.
- Dodanie losowości lub asymetryczne zmienianie kolejności zabierania widelców mogą zmniejszyć szanse na zakleszczenie.
- Wykorzystując np. arbitra (lokaja), który ogranicza liczbę filozofów przy stole, czas oczekiwania staje się bardziej przewidywalny. Może to poprawić równomierny dostęp do widelców.
- W miarę zwiększania liczby filozofów, konkurencja o widelce staje się większa, co wydłuża czasy oczekiwania.