



Teoria współbieżności

LABORATORIUM 3
PROBLEM PIĘCIU FILOZOFÓW

JAKUB RADEK

03.11.2022

1 Opis problemu

Problem pięciu filozofów jest jednym z klasycznych problemów teorii współbieżności. Podstawowe sformułowanie problemu jest następujące :

- N filozofów zasiada przy okrągłym stole
- Pomiedzy sąsiednimi filozofami leży widelec (łącznie jest N widelców)
- Każdy filozof działa ciągle według schematu „myślenie - jedzenie - myślenie - jedzenie - ...”. Każdy z etapów (myślenie i jedzenie) jest skończony.
- Aby zjeść, filozof musi podnieść oba sąsiadujące widelce

Zadanie: zaprojektuj algorytm jednoczesnej alokacji współdzielonych zasobów (widelce) przez konkurujące procesy (filozofowie), tak aby uniknąć zakleszczenia i zagłodzenia. Niektóre z rozwiązań problemu pięciu filozofów są następujące.

- **Rozwiązanie naiwne (z możliwością blokady).** Każdy filozof czeka, aż wolny będzie lewy widelec, a następnie go podnosi (zajmuje), następnie podobnie postępuje z prawym widelcem.
- **Rozwiązanie z możliwością zagłodzenia.** Każdy filozof sprawdza czy oba sąsiednie widelce są wolne i dopiero wtedy zajmuje je jednocześnie. Rozwiązanie to jest wolne od blokady, jednak w przypadku, gdy zawsze któryś z sąsiadów będzie zajęty jedzeniem, nastąpi zagłodzenie, gdyż oba widelce nigdy nie będą wolne.
- **Rozwiązanie asymetryczne.** Filozofowie są ponumerowani. Filozof z parzystym numerem najpierw podnosi prawy widelec, filozof z nieparzystym numerem najpierw podnosi lewy widelec.
- **Rozwiązanie z arbitrem.** Zewnętrzny arbiter (łokaj, kelner) pilnuje, aby jednocześnie co najwyżej czterech (w ogólnym przypadku $N-1$) filozofów konkurowało o widelce. Jeśli naraz wszyscy filozofowie będą chcieli jeść, arbiter powstrzymuje jednego z nich aż do czasu, gdy któryś z filozofów skończy jeść.

2 Implementacja rozwiązań

1. Java

Rozwiązanie problemu dla wariantu z możliwością zagłodzenia oraz dla wariantu z arbitrem zostały zaimplementowane w języku Java.

2. JavaScript

Rozwiązania problemu dla wszystkich czterech podanych wariantów zostały zaimplementowane w języku JavaScript.

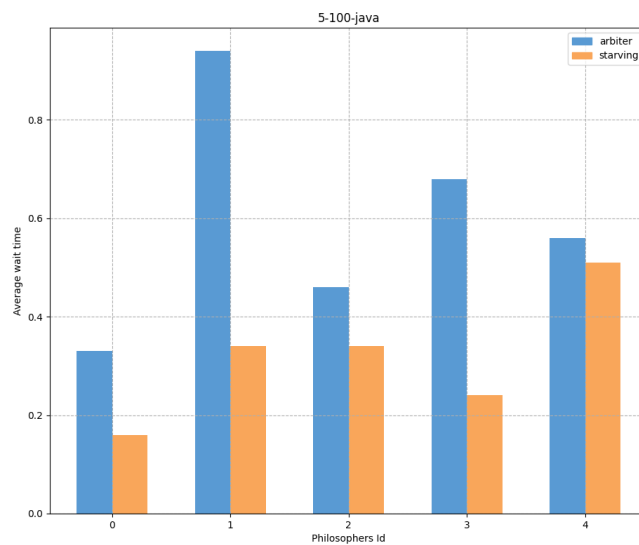
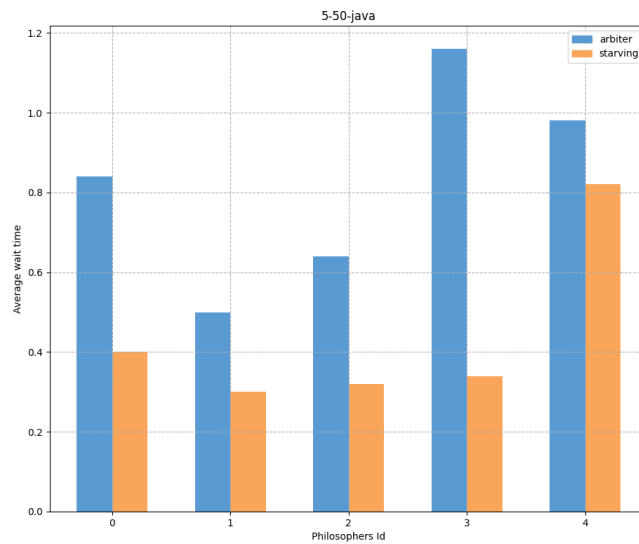
3 Opracowanie wyników

Zaimplementowane rozwiązania zostały wykorzystane do uzyskania paru zestawów danych, gdzie mierzony był średni czas oczekiwania każdego z filozofów na dostęp do jedzenia.

Wyniki zostały pogrupowane na podstawie ilości filozofów oraz ilości iteracji, uzyskując wykresy słupkowe gdzie kolor odpowiada metodzie rozwiązania problemu.

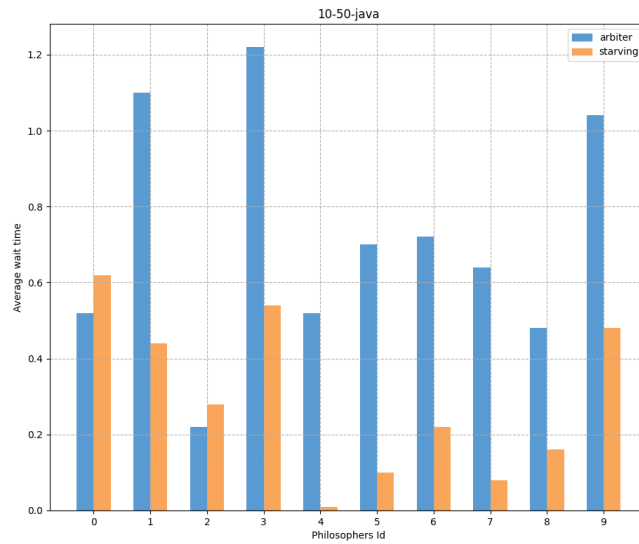
Ze względu na możliwość zakleszczenia się programu dla metody naiwnej, została ona pominięta w wynikach.

3.1 Java

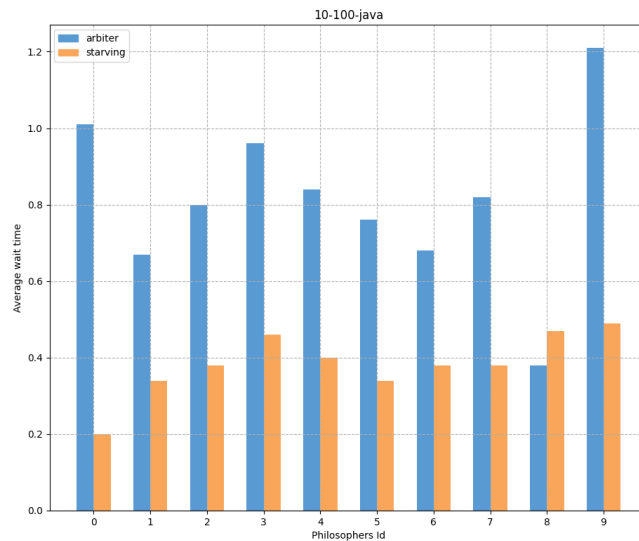


Powyższe wykresy ilustrują trend związany z dłuższym czasem oczekiwania na posiłek dla rozwiązania stosującego arbitra, zazwyczaj jeden z filozofów ma znacząco dłuższy czas oczekiwania od pozostałych.

Rozwiązanie z możliwością zagłodzenia uzyskuje lepsze(krótsze) średnie czasy oczekiwania na posiłek.



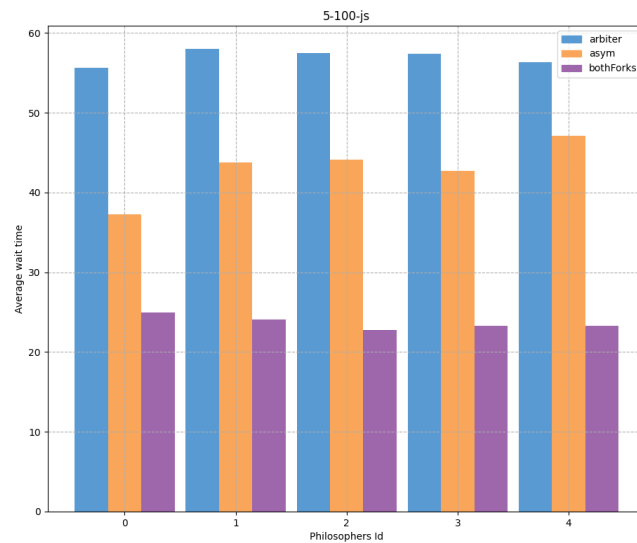
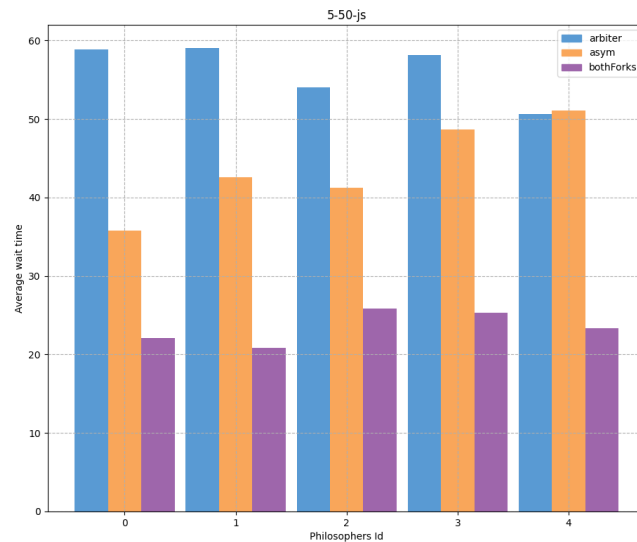
Przypadek powyżej ilustruje nieprzewidywalność rozwiązania z możliwością zagłodzenia, uzyskana została bowiem sytuacja, gdy jeden z filozofów ma prawie zerowy czas oczekiwania.



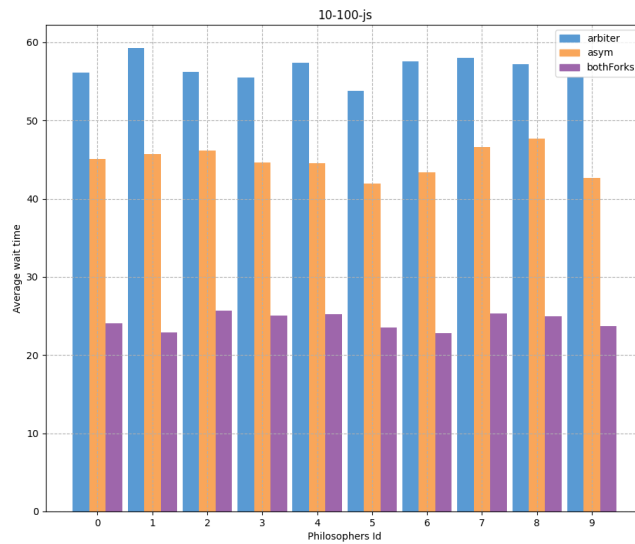
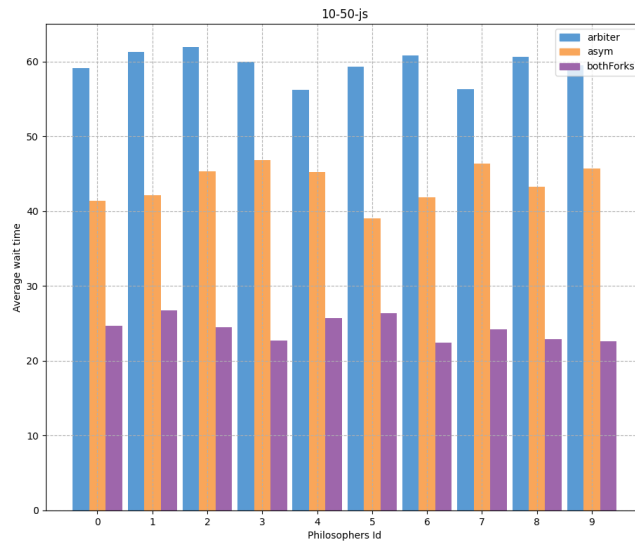
Uzyskany wykres ilustruje przypadek dla którego średnie czasy oczekiwania dla każdej z metod dla każdego z filozofów są dość podobne, występuje natomiast pojedyńczy filozof ze znacznie krótszym czasem oczekiwania dla metody z arbitrem.

Porównując wykresy można zaobserwować, że rozwiązanie z możliwością zagłodzenia jest dla danego przypadku i danych wydajniejsze od rozwiązania stosującego arbitra.

3.2 JavaScript



Powyższe wykresy uzyskują bardzo podobne wyniki pomimo zwiększonej liczby iteracji. Można zauważyć trend sugerujący wydajność rozwiązań, gdzie najwydajniejszym rozwiązaniem jest rozwiązanie z jednoczesnym podnoszeniem obu widelców, a najmniej wydajne rozwiązanie z arbitrem. Występuje jeden przypadek gdzie rozwiązanie asymetryczne uzyskało gorszy średni czas dostępu od rozwiązania stosującego arbitra.



Wykresy dla większej ilości filozofów utwierdzają w przekonaniu o wydajności każdego z rozwiązań.

Na podstawie uzyskanych wyników można stwierdzić, że najwydajniejszą implementacją dla problemu filozofów jest metoda oparta o jednoczesne podnoszenie widelców, uzyskuje ona stale lepsze wyniki od pozostałych metod.

3.3 Porównanie między językami

Wyniki uzyskane przy pomocy implementacji w języku Java uzyskują zazwyczaj wyższe odchylenie standardowe, wyniki uzyskane implementacjami w języku JavaScript dają znacznie bardziej podobne do siebie wyniki.

Niezależnie od języka najwydajniejszym sposobem implementacji wydaje się być sposób z możliwością zagłódnienia (jednoczesne podnoszenie widelców), natomiast dla implementacji w języku Java rozwiązanie to jest bardzo nieprzewidywalne. Natomiast niezależnie od języka rozwiązanie to jest mniej zbalansowane, pomimo lepszych średnich czasów dostępu doprowadza ono do sytuacji gdzie występują przypadki prawie zerowego czasu oczekiwania, bądź znacząco wydłużonego czasu oczekiwania, dlatego nie powinno się go stosować.