

# Article template

May 27, 2025

**Adrian Herda**

Informatyka Algorytmiczna, Politechnika Wrocławska

## 1. Algorytm

### 1.1. Cel

Ten algorytm zarządza kopiami stron w systemie, gdzie operacje odczytu i zapisu mają różny koszt zależnie od liczby dostępnych kopii danej strony. Głównym celem algorytmu jest zoptymalizowanie kosztów dostępu do stron poprzez inteligentne tworzenie i usuwanie ich kopii na podstawie częstotliwości użycia.

### 1.2. Działanie

#### 1.2.1. Początek

Na początku system ma jedną kopię strony 0, a jej stan zostaje ustawiony na „oczekujący” (Waiting). Dla każdej z 64 stron system przechowuje licznik użyc (czyli ile razy strona była czytana lub pisana) oraz stan logiczny informujący, czy strona jest w trybie normalnym czy oczekującym. Dodatkowo przechowywany jest zbiór aktualnie posiadanych kopii stron.

#### 1.2.2. Operacja read

Podczas odczytu strony, jeśli strona nie ma kopii, algorytm dolicza koszt (czyli symuluje, że trzeba ją pobrać) i zwiększa licznik jej użycia. Jeśli licznik osiąga wcześniej zdefiniowany próg (threshold), tworzona jest nowa kopia tej strony. Stworzenie nowej kopii również wiąże się z kosztem równym progowi.

#### 1.2.3. Operacja write

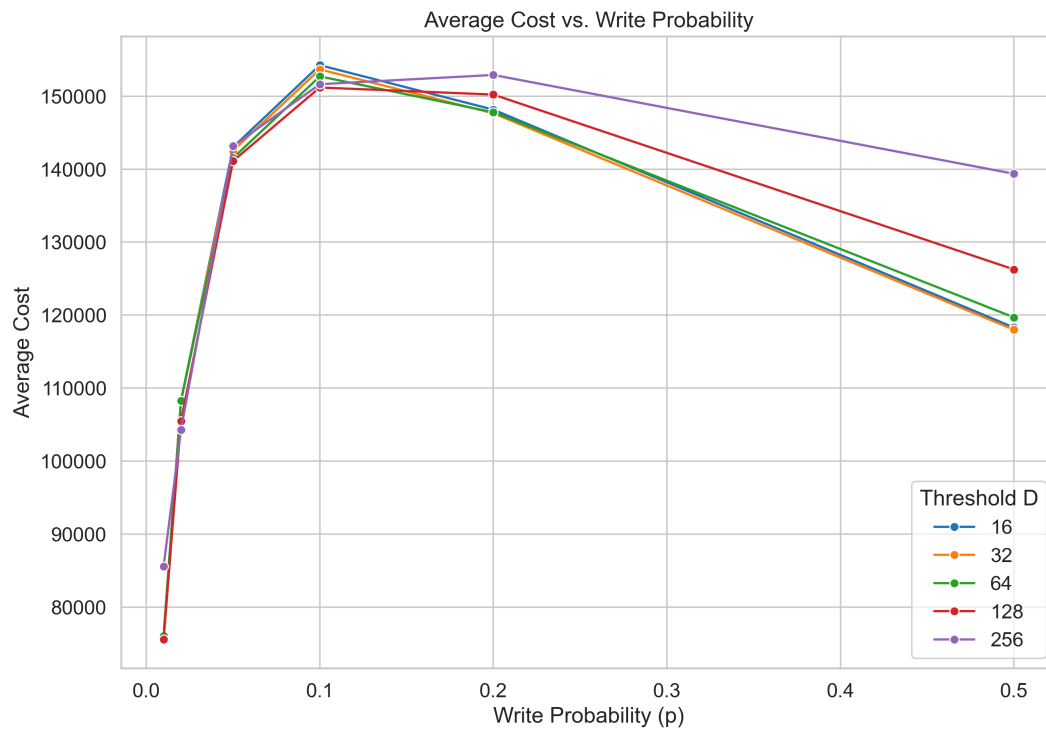
Podczas zapisu algorytm działa inaczej. Jeśli strona ma już kopię, koszt zapisu zależy od liczby pozostałych kopii (każdą trzeba zaktualizować, więc koszt to liczba kopii minus jeden). Jeśli strona nie ma kopii, trzeba ją zsynchronizować ze wszystkimi pozostałymi kopiami, co oznacza koszt równy liczbie wszystkich kopii. W niektórych przypadkach (gdy jest tylko jedna kopia w systemie i jakaś strona oczekuje) można inkrementować licznik tej strony. Po tym zapisie system dodatkowo aktualizuje wszystkie inne strony — jeśli któraś z nich ma kopię i licznik większy od zera, to licznik jest zmniejszany. Jeśli licznik spadnie do zera, system usuwa kopię strony (chyba że to jedyna pozostała kopia – wtedy pozostaje w stanie oczekującym).

#### 1.2.4. Tworzenie kopii

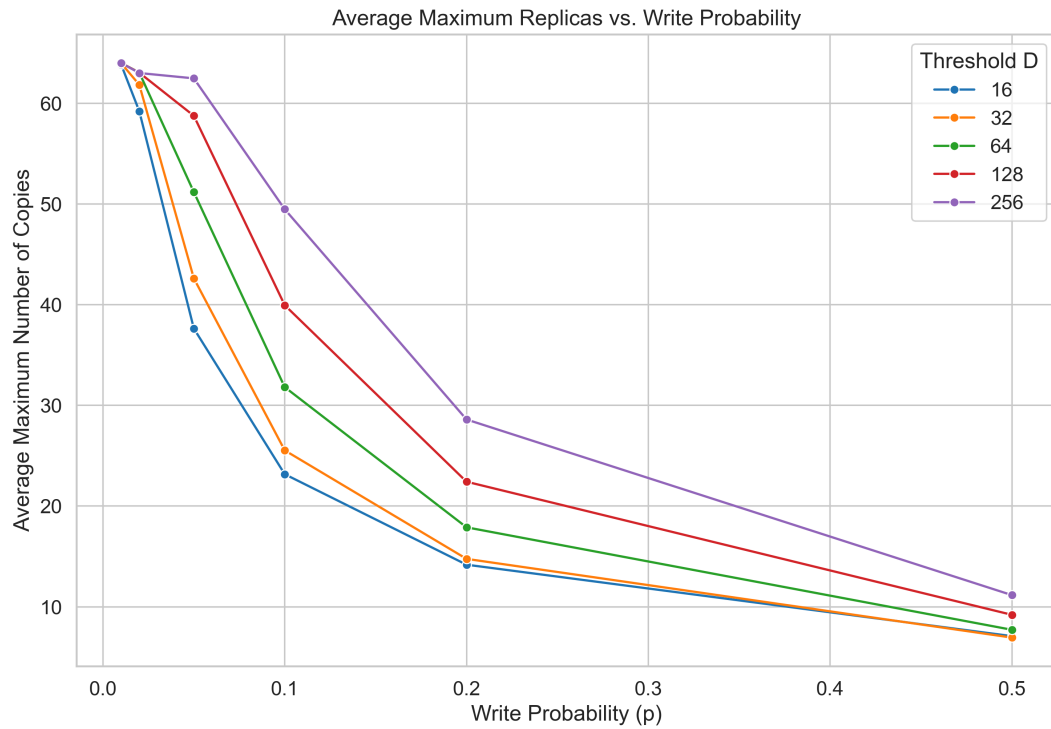
Tworzenie kopii strony następuje tylko wtedy, gdy licznik użycia osiąga próg. Kopia zostaje dodana, a jeśli któraś ze stron była w stanie oczekującym, to jedna z nich zostaje usunięta, aby kontrolować liczbę kopii. System śledzi także największą liczbę kopii, jaka wystąpiła w czasie działania (max\_copies).

Całość symuluje zachowanie systemu, który adaptacyjnie replikując dane (strony) stara się zredukować koszty operacji w zależności od ich typu i częstotliwości. Odczyty premią tworzenie kopii, natomiast zapisy powodują ich usuwanie lub synchronizację. To odzwierciedla rzeczywiste strategie w systemach rozproszonych i bazach danych, gdzie trzeba balansować między kosztownymi operacjami zapisu a przyspieszaniem odczytów przez replikację.

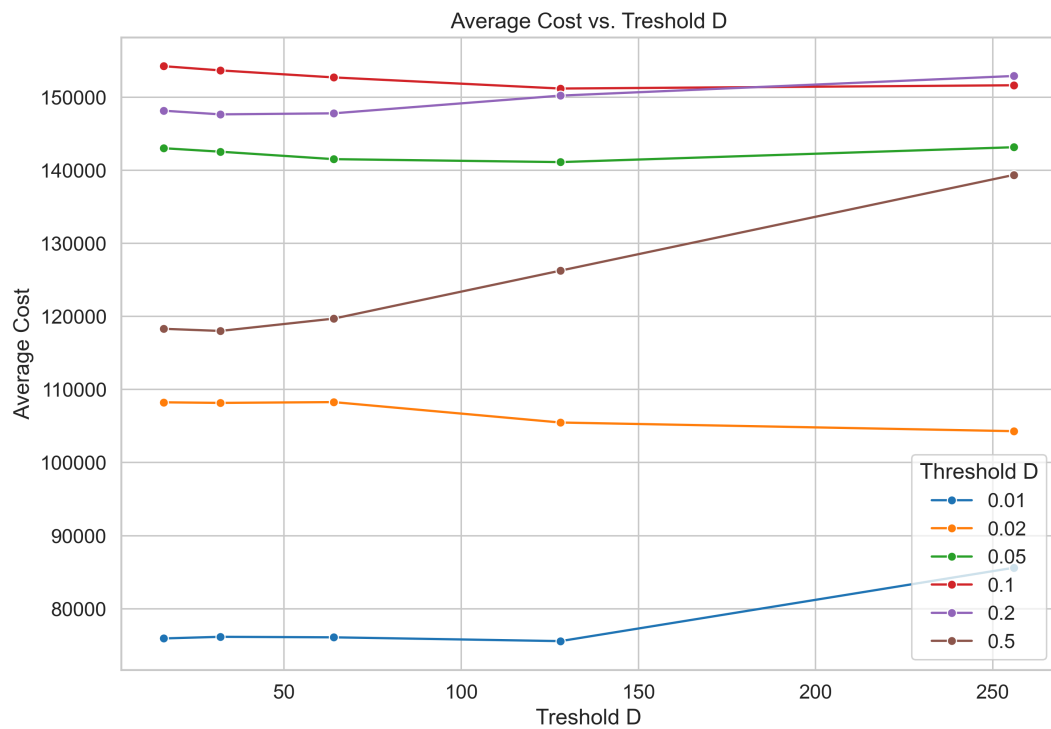
## 2. Wykresy



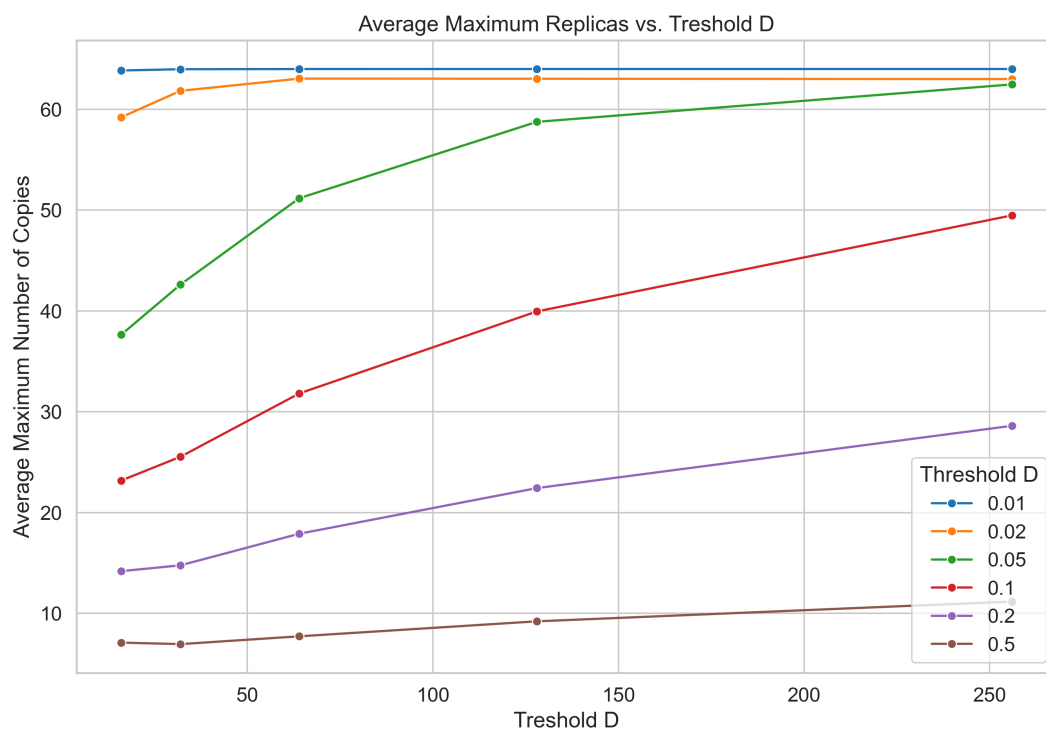
Rysunek 1: Średni koszt operacji w zależności od prawdopodobieństwa zapisywania (p)



Rysunek 2: Średnia maksymalna ilość w zależności od prawdopodobieństwa zapisywania ( $p$ )



Rysunek 3: Średni koszt operacji w zależności od kosztu kopii zasobu ( $D$ )



Rysunek 4: Średnia maksymalna ilość w zależności od kosztu kopii zasobu (D)