

POŚREDNIKI SPAMIĘTUJĄCE — ALGORYTMY  
WEWNĘTRZNE

1. WSTĘP

A) SPAMIĘTYWANIE PAMIĘCI OPERACYJNEJ

B) SPAMIĘTYWANIE W BAZACH DANYCH

C) SPAMIĘTYWANIE W ROZPROSZONYCH SYSTEMACH PLIKÓW

**AD A)** RÓŻNICE MIĘDZY PAMIĘCIĄ OPERACYJNĄ A WWW (CECHY WWW):

WWW — RÓŻNY ROZMIAR OBIEKTÓW, (DLA WWW)

WWW — ZMIENIAJĄCY SIĘ ROZMIAR OBIEKTÓW, (...)

WWW — RÓŻNY CZAS SPROWADZANIA RÓŻNYCH OBIEKTÓW (NAWET TEGO SAMEGO ROZM.)

WWW — NIĘKSZE MOŻLIWOŚĆ OBLICZENIOWE

**AD B)** RÓŻNICE MIĘDZY BAZAMI DANYCH A WWW (Z PUNKTU WIDZENIA WWW)

WWW — BRAK PRIORYTETÓW DANYCH,

WWW — BRAK MOŻLIWOŚCI WYKONYWANIA ANALIZY STATYCZNEJ,

WWW — RÓŻNY ROZMIAR OBIEKTÓW

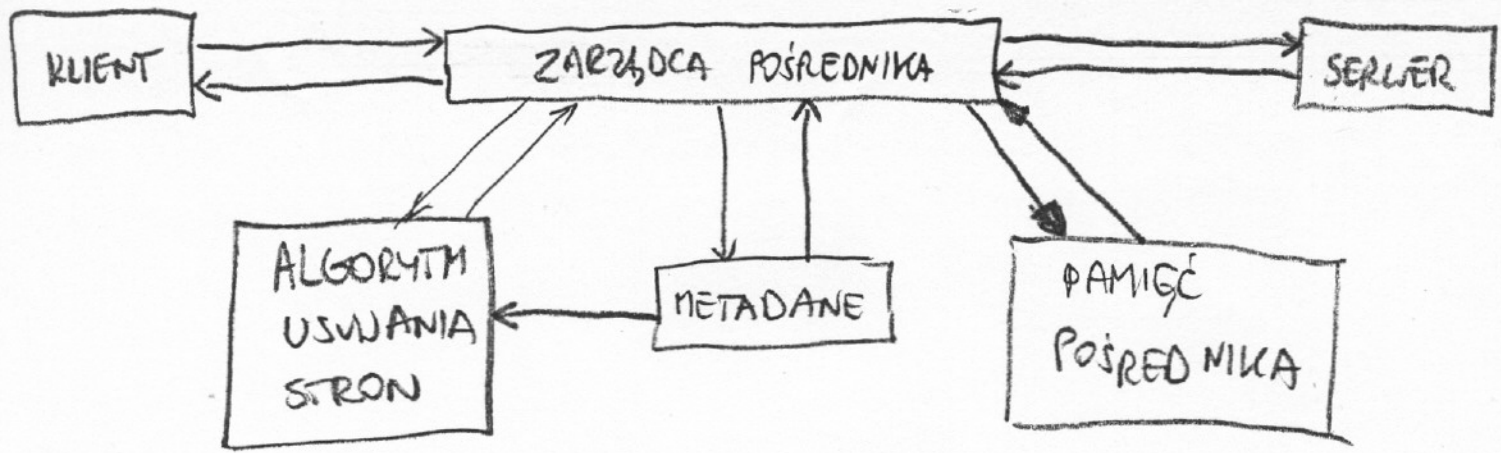
WWW — RÓŻNY CZAS SPROWADZANIA RÓŻNYCH OBIEKTÓW (NAWET TEGO SAMEGO ROZMIARU)

**AD C)** RÓŻNICE MIĘDZY SYSTEMAMI PLIKÓW A WWW

WWW — RÓŻNY CZAS SPROWADZANIA RÓŻNYCH OBIEKTÓW (NAWET TEGO SAMEGO ROZMIARU)

NADWAŻNIWSZA RÓŻNICA: BRAK ZAPISU W TYM SAMYM  
KANALE DANYCH (DLA WWW)

## 2. BUDOWA POŚREDNIKA SPAMIĘTUJĄCEGO



## 3. ALGORYTMY USUWANIA STRON — PRZEGLĄD

### • OPT (OPTIMAL)

ALGORYTM OPTIMALNY — NIEMOŻLIWY W REALIZACJI, BO WYMAGA ZNACZOMOŚCI CAŁEGO CIĄGU ZLECEŃ — NORMALNIE NIE MAMY DOSTĘPU DO ZLECEŃ PRZYSZŁYCH. UŻYWAMY PRZY TEORETYCZNYCH PORÓWNIANIACH.

### • NRU (NOT RECENTLY USED)

MAMY DWA BITY R ORAZ M. PRZY ODCZYTANIE  $R := 1$ , PRZY ZAPISIE  $M := 1$ . OKRESIŁO R I M SĄ CZYSZCZONE. USUWANA JEST STRONA Z  $R=0$ . JEŚLI JEST KILKA TAKICH STRON, TO PREFEROWANE DO USUNIĘCIA SĄ STRONY Z  $M=0$ .

ZALETY: PROSTOTA, DOBRA EFEKTYWNOŚĆ. STOSOWANY W ZARZĄDZANIU PAMIĘCIĄ OPERACYJNĄ.

### • FIFO (FIRST IN, FIRST-OUT)

USUWANA JEST STRONA NAJDDURZEDZIEJ ZNAJDUJĄCA SIĘ W PAMIĘCI. STOSOWANE W SYTUACJI, GDY DOSTĘP DO STRON JEST SEKWENCYJNY.

• LRU (LEAST RECENTLY USED)

PRZYBUZENIE OPT OPARTE NA ZAŁOŻENIU LOKALNOŚCI ODWOŁAŃ. DOKUMENTY CZĘSTO WYUWANE W NIEDAWNEJ PRZESZŁOŚCI BĘDĄ PRAWDOPODOBNIEMIE CZĘSTO WYUWANE W NIEDALEKIEJ PRZYSZŁOŚCI. W TYM ALGORYTMIE USUWANE SĄ DOKUMENTY, KTÓRE NIE BYŁY WYUWANE PRZEZ NADŁUŻYSZY OKRES AŻ DO MOMENTU ZEDSIA POMIĘDZ PEWNEGO PROGU.

• LFU (LEAST FREQUENTLY USED)

PRZY TYM PODEJŚCIU ZLICZA SIĘ DOSTĘPY DO STRONY. Z PAMIĘCI USUWANA JEST STRONA O NAJMNIEDSZYM LICZNIKU. WADA: PEWNE STRONY SĄ POPULARNE TYLKO W KRÓTKIM ODCINKU CZASU. W TYM OKRESIE NABYDADĄ LICZNIK, A POTEM JEDNOCZEŚNIE NIE SĄ WYUWANE ORAZ ZAJMADĄ MIEJSCE W POŚREDNIKU.

• SIZE

Z PAMIĘCI USUWANY JEST NADWIĘKSZY DOKUMENT. TA STRATEGIA POZWALA OSIĄGNĄĆ WYSOKIE ILOŚCI TRAFIENI, JEDNAK DZIEJE SIĘ TO KOSZTEM ILOŚCI DANYCH W TRAFIENIACH. INNA WADA POLEGA NA ZAŚMIECANIU PAMIĘCI MAŁYMI STRONAMI.

• LRU-THRESHOLD

DZIAŁA JAK LRU Z TYM, ŻE NIE ZAPAMIĘTUJE DOKUMENTÓW POWIĘD OKREŚLONEGO, KONFIGUROWALNEGO ROZMIARU. W PRZYPADKU ZAPEŁNIONEJ PAMIĘCI

• LRU-MIN

UZYWAMY LRU, ALE DLA KOLEJNYCH PODZBIORÓW DOKUMENTÓW ZNADDDAJĄCYCH SIĘ W PAMIĘCI PODZBIORY GENEROWANE SĄ LG REGUŁY: NADPĘD DOKUMENTY > PROGU S, POTEM > OD PROGU S/2, POTEM > S/4 ITD. JEST TO SAMOSTERUJĄCA SIĘ WERSJA LRU-THRESHOLD.



9

## • HYPER-G

SORTUJEMY DOKUMENTY LEKSYKOGRAFICZNIE WEDŁUG TRÓJKI  
(LICZBA ODWOŁAŃ, - CZAS OD OST. ODWOŁANIA, - ROZMIAR)

USUWAMY ODPOWIEDNIĄ ILOŚĆ DOKUMENTÓW O NAJMNIEJSZYCH  
WARTOŚCIACH POWRZESZED TRÓJKI.

## • PITKOW / RECHER

STOSUJEMY LRU Z DODATKOWYM PROGIEM; JEŚLI WSZYSTKIE  
DOKUMENTY BYŁY ŚCIAĞANE TEGO SAMEGO DNIA, TO USUWAMY  
NADWISZKIE

## • LLF (LOWEST LATENCY FIRST)

DLA KAZDEGO DOKUMENTU ZAPISUJEMY CZAS ŚCIAĞANIA.  
USUWAMY DOKUMENTY O NAJMNIEJSZYM CZASIE ŚCIAĞANIA.

NADA: TA STRATEGIA NIE MINIMALIZUJE ŚREDNIEGO  
CZASU OCZEKIWANIA.

## • HYBRID

DLA KAZDEGO DOKUMENTU OBLICZANA JEST  
WARTOŚĆ UŻYTECZNOŚCI:

$$\frac{\left(c_s + \frac{w_b}{b_s}\right) n_p^{w_h}}{z_p}$$

GDZIE:

- $c_s$  - CZAS ŁĄCZENIA SIĘ Z SERWIEREM S
- $b_s$  - PRZEPYŃKOWOŚĆ DO SERWERA S
- $n_p$  - LICZBA ODWOŁAŃ DO STRONY OD MOMENTU JEJ ŚCIAĞNIĘCIA
- $z_p$  - ROZMIAR DOKUMENTU P W BAJTACH
- $w_b, w_h$  - ODPOWIEDNIO WYBRANE STALE.

HYBRID DAJE LEPSZY ŚREDNI CZAS DOSTĘPU NIŻ LLF.

(5)

ZA TO JEST NIEPRAKTYCZNY - TRUDNO OBLICZALNE PARAMETRY.

#### 4. ALGORYTMY STOSOWANE W SQUIDZIE

A) LRU

B) GDSF

C) LFU-DA

AD A) ODRZYWIŚTA STRATEGIA

AD B) GDSF: GREEDY-DUAL SIZE FREQUENCY

#### • PRZYBLIŻENIE I: GREEDY-DUAL (YOUNG)

- JEDNOLITY ROZMIAR
  - RÓŻNE CZASY ŚCIĄGANIA
- } WTEDY DOBRZE DZIAŁA

ZASADA DZIAŁANIA: + KAŻDA STRONA MA PRZYPISANY KLUCZ  $K(p)$

+ PO ŚCIĄGNIĘCIU  $K(p) =$  CZAS ŚCIĄGNIĘCIA STRONY

OPTIMALIZACJA  
ZA POMOCĄ  
"RUCHOMEGO ZERA"  
 $K(p) = L + \text{CZAS ŚCIĄGANIA}$

- + PRZY USUWANIU ZNAJDOwana JEST STRONA
- + O MINIMALNYM  $K(p)$  RÓWNYM min
- + USUWAMY  $p$
- + OD  $K(p)$  KAŻDEJ STRONY ODEJMOWANE  
JEST min (\* kosztowne dlatego stosujemy optymalizację "ruchomego zera")
- + PRZY ODWOŁANIU DO STRONY  $p'$   
PRZYWRACANE JEST  $K(p)$  RÓWNE CZASOWI ŚCIĄGANIA STRONY

#### • PRZYBLIŻENIE II: GREEDY-DUAL SIZE (CAO & IRANI)

+ J.W. ALE  $K(p) = \frac{\text{CZAS ŚCIĄGANIA STRONY}}{\text{ROZMIAR STRONY}}$

DZIĘKI TEMU DAJEMY SOBIE RADĘ Z RÓŻNYMI ROZMIARAMI

• PROXIMIZEME III : GDSF

6

+ JAK GD, ALE

$$K(p) = \text{CZĘSTOTLIWOŚĆ ODWOŁAŃ} * \frac{\text{CZAS ŚLĄGAMA STRONY}}{\text{ROZMIAR STRONY}}$$

+ CZĘSTOTLIWOŚĆ ODWOŁAŃ JEST INKREMENTOWANA PRZY KAŻDYM ODWOŁANIU

c) LFU-DA : LEAST FREQUENTLY USED WITH DYNAMIC AGING

+ JAK GD, ALE

$$K(p) = \text{CZĘSTOTLIWOŚĆ ODWOŁAŃ}$$

(DYNAMIC AGING — OPTIMALIZACJA ZA POMOCĄ „RUCHOMEGO ZERA”)

5. WAŻNA UWAGA:

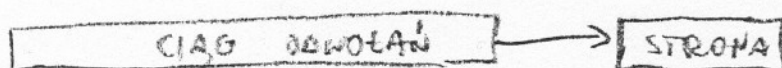
OBRAZKI, DŹWIĘKI ITP. ZE STRONY SĄ PRAWIE ZAWSZE ŚLĄGANE ZE STRONĄ — OPŁACA SIĘ WPROWADZIĆ MECHANIZMY TRAKTOWANIA ICH ŁĄCZNE.

POMYSŁ: WPROWADZIĆ HEURYSTYKI — OBRAZKI ŚLĄGANE SĄ MIĘDZY DOKUMENTAMI HTML.

POWZESIE TRZEBA POPRAWIAĆ PRZY KAŻDYM ODWOŁANIU: MOGŁO BYĆ ODWOŁANIE ZANIM WSZYŚKIE OBRAZKI SIĘ ŚCIAGNĘŁY

6. WYRAFINOWANE HEURYSTYKI

STARAMY SIĘ ŚLEDZIĆ SESJE. BUDUJEMY WZORCE POSTACI



PRZYMASZĄC STRZELKOM PRAWDOPODOBIEŃSTWA, WSUWAMY  
STRONĘ O MINIMALNYM PRAWDOPODOBIEŃSTWIE. (TO TRZEBA  
ODPOWIEDNIO WKLUCZYĆ, BO CIĄGI NOGĄ, MIEĆ RÓŻNE DŁUGOŚCI).

7