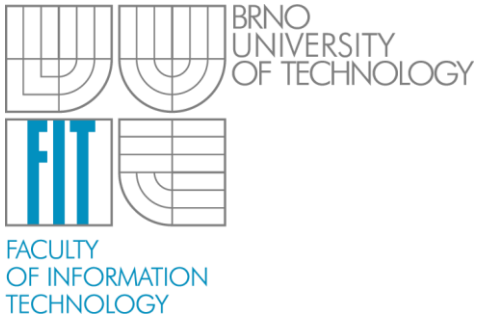
Správy cache

Martin Žádník

Vysoké učení technické v Brně, Fakulta informačních technologií v Brně
Božetěchova 2, 612 66 Brno
ant@fit.vutbr.cz

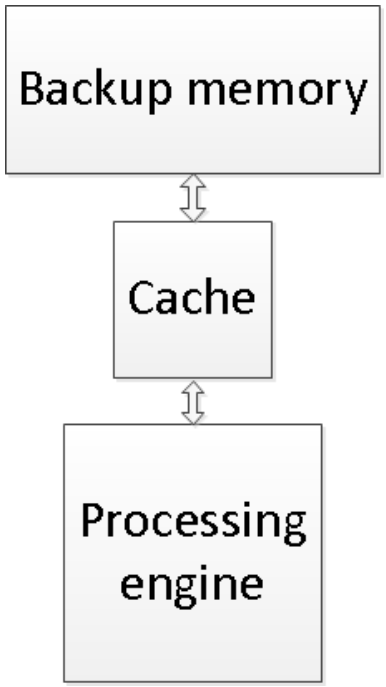


INVESTMENTS IN EDUCATION DEVELOPMENT

Hierarchie

- Cílem cache je dostat data co nejblíže výpočetnímu prvku

- Nejčastěji hierarchie pamětí
  - Cache
  - Hlavní paměť



- Ano či ne?
  - Technologie vs. Režie
  - Zrychlení vs. Zpoždění

Motivace

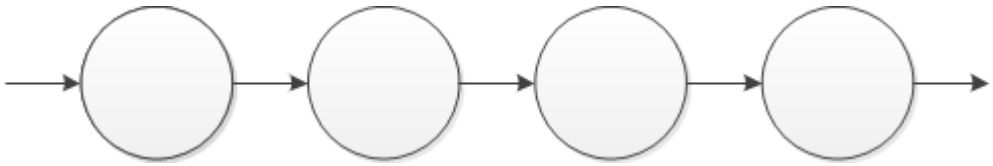
- Cache procesoru
  - Instrukční, datová
  - L1, L2
- Cache překladu stránek
- Databázová cache
- Web cache
- Flow cache
- Distribuovaná cache obsahu

Problém

- Správa cache
  - Zvýšit úspěšnost vyhledání položky v cache
- Volba správy závisí
  - Nejvíce na datech
  - Výpočetní a paměťové náročnosti správy
  - Velikosti paměti

Triviální

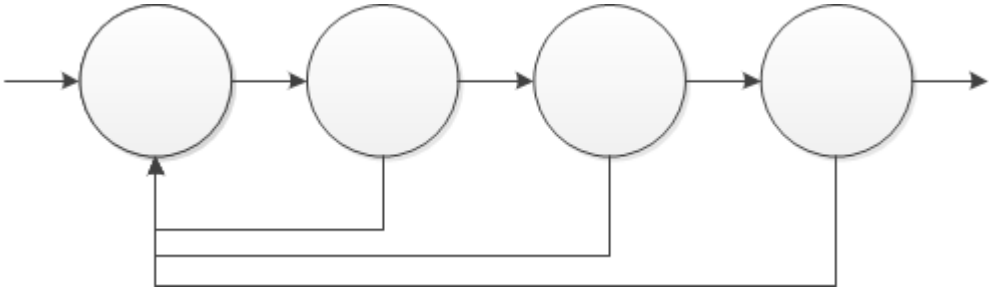
- FIFO
  - Seznam položek
  - Nové položky na začátek seznamu
  - Při nedostatku místa odebírám z konce seznamu
  - Vhodné pro položky s omezenou a stejnou dobou života
  - Přidat, odebrat je O(1), vyhledání O(n)?



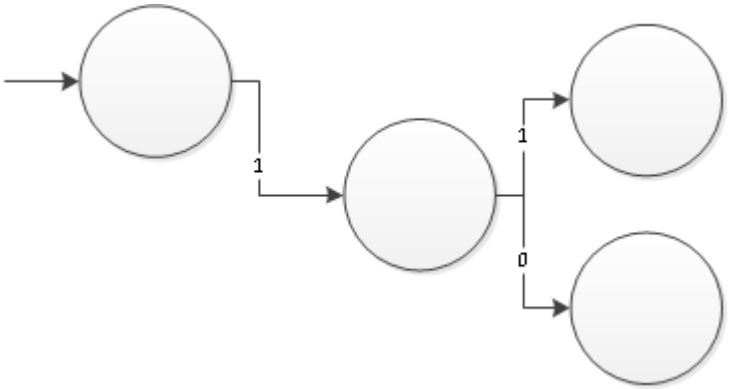
- Random
  - Aproximace FIFO
  - Čím déle v cache, tím větší pravděpodobnost odstr.
  - Nulová režie

Nedávnost přístupů

- LRU – least recently used
  - Seznam položek
  - Nové položky na začátek seznamu
  - Při nedostatku místa odebírám z konce seznamu
  - Vhodné téměř pro všechny typy dat s lokalitou
  - Přidat, odebrat je O(1), vyhledání O(n)?

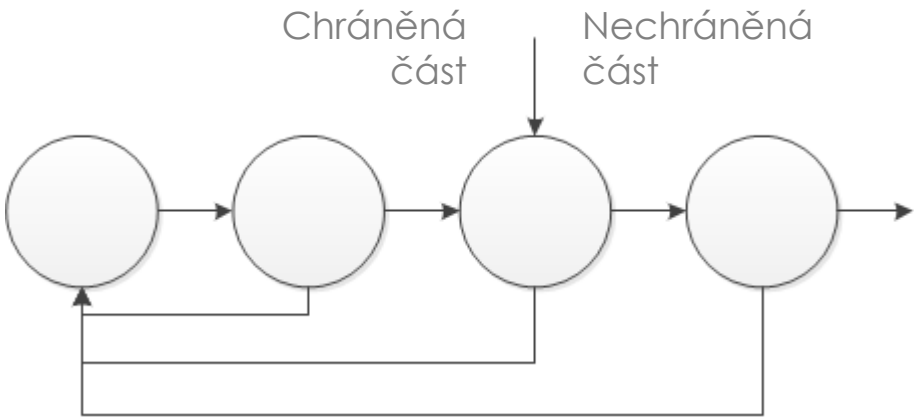


- Pseudo-LRU
  - Implementace pomocí stromu



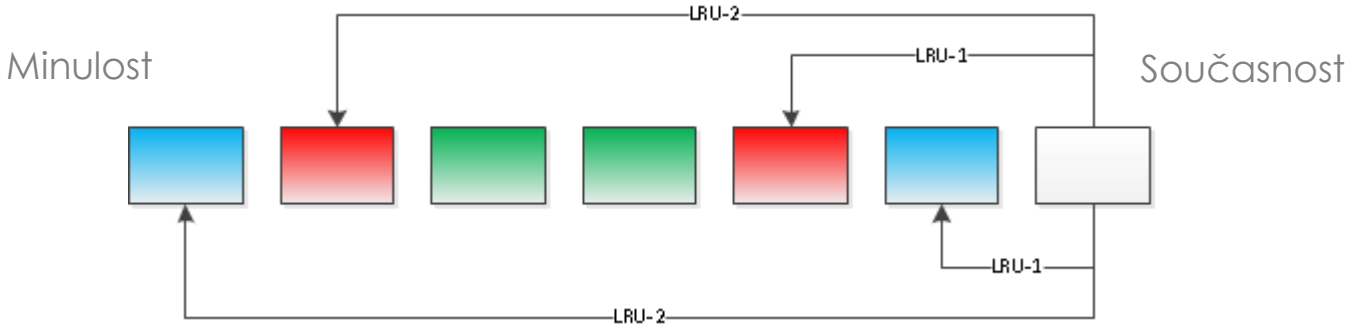
Nedávnost přístupů

- Segmented LRU
  - Rozdělení seznamu na 2 části
  - Rozdělení pozicí vkládání nových položek
  - Chráněná a nechráněná část



Nedávnost přístupů

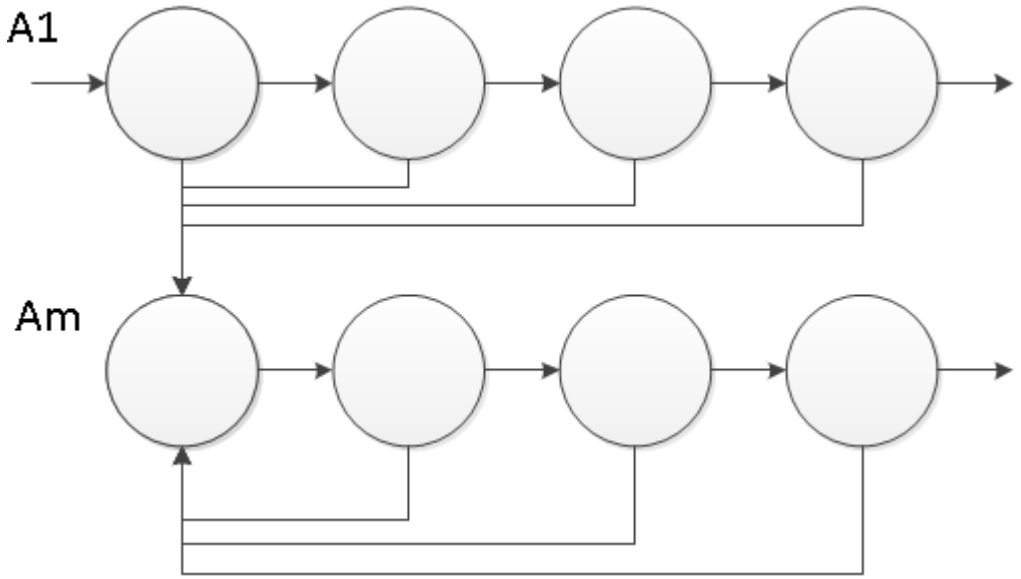
- LRU-K především LRU-2
  - Uvažuje nejméně používanou položku
  - Zpětně „K“ přístupů do historie
  - Pokud nemá položka „K“ přístupů, pak INF
  - INF spravované jako LRU
  - Příchody požadavků viz obrázek



  - Složitá implementace řadícím stromem

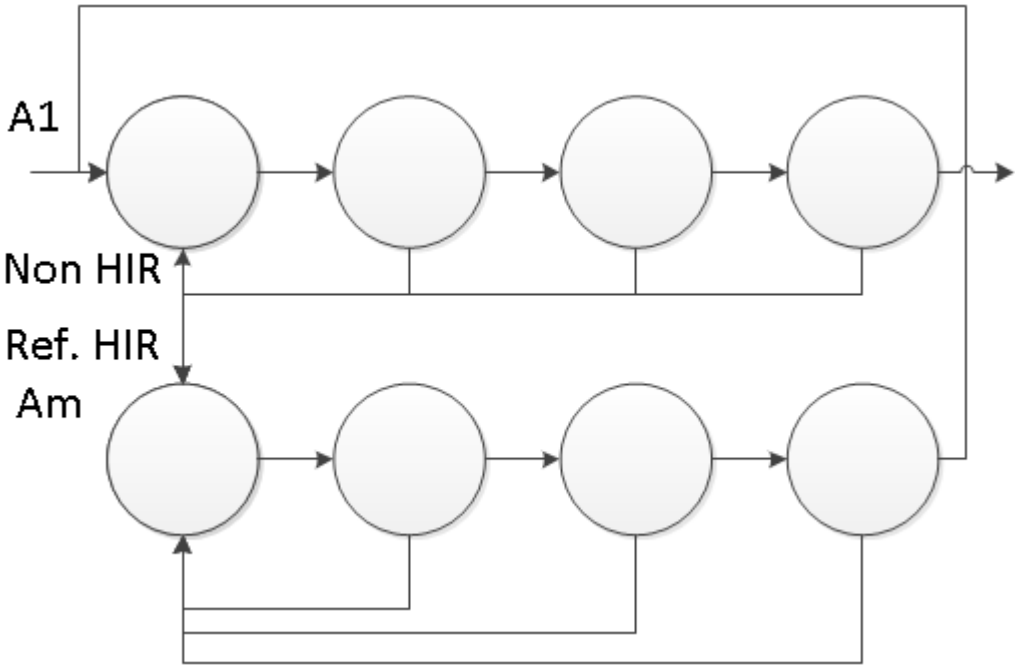
Nedávnost přístupů

- 2Q
  - Aproximace LRU-2 pomocí 2 seznamů A_m, A_1
  - A_1 spravováno jako FIFO při hit přesun do A_m
  - A_m spravováno jako LRU



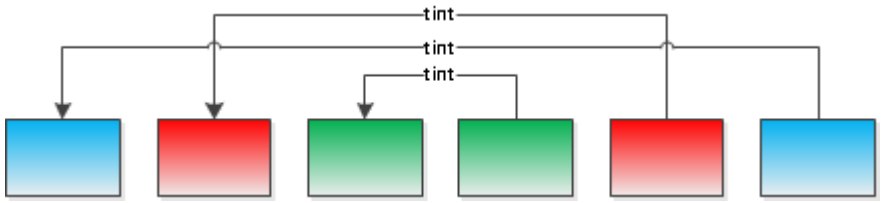
Nedávnost přístupů

- LIRS
  - Low inter-reference recency
  - LIR vs. HIR blocks
  - 2 LRU seznamy
  - Zavedení resident HIR a non-resident HIR



Nedávnost přístupů

- Exponential smoothing
  - Váhovaný klouzavý průměr intervalů mezi přístupy
  - S příchodem paketu je vykonána následující funkce
  - $W = aW + (1-a)t_{int}$



  - Problém se stárnutím položek
  - Problém s vyhledáním nejstarší – řadící strom

Frekvence přístupů

- LFU – least frequently used
  - Sleduje se četnost přístupů k položkám
  - Nejméně používaná je odstraněna
  - Problém: stárnutí a přílišné popularizace
- LFU*-aging
  - Limit pro čítač počtu přístupů
  - Aging – jednou za čas snížení všech čítačů na polovinu

Frekvence přístupů

- LFU – DA = LFU with dynamic aging
  - Zavádí faktor běžícího *L*.
  - Faktor *L* má na začátku hodnotu nula, a v průběhu uvolňování položek je aktualizován na $L=K_i$, kde K_i je prioritní klíč uvolněné položky i, který se spočítá jako:
  - $K_i = F_i + L_i$,
  - kde F_i je počet přístupů k položce i. Hodnota L_i je uložena do každé položky v okamžiku jejího vložení do cache a je rovna hodnotě *L* v daném okamžiku.
  - Rozhodnutí, které položky uvolnit je prováděno na základě prioritního klíče K_i a nikoliv pouze na základě frekvence F_i.

Adaptivní správy

- SEQ
  - detekce sekvenčních přístupů ke stránkám
- DEAR – pokračování SEQ
  - klasifikace přístupů sekv., cyklický, shluk., náhodný
- UBM – pokračování DEAR
- EELRU – při zvýšeném počtu výpadků přepnout z LRU na MRU

Závěr

- Dle mých experimentů na síťových tocích pracuje nejlépe SLRU
- Pro správu web cache apod. je vhodná LFU*-aging
- Pro správu cache procesoru LRU, LIRS nebo adaptivní

Konec

Děkuji za pozornost