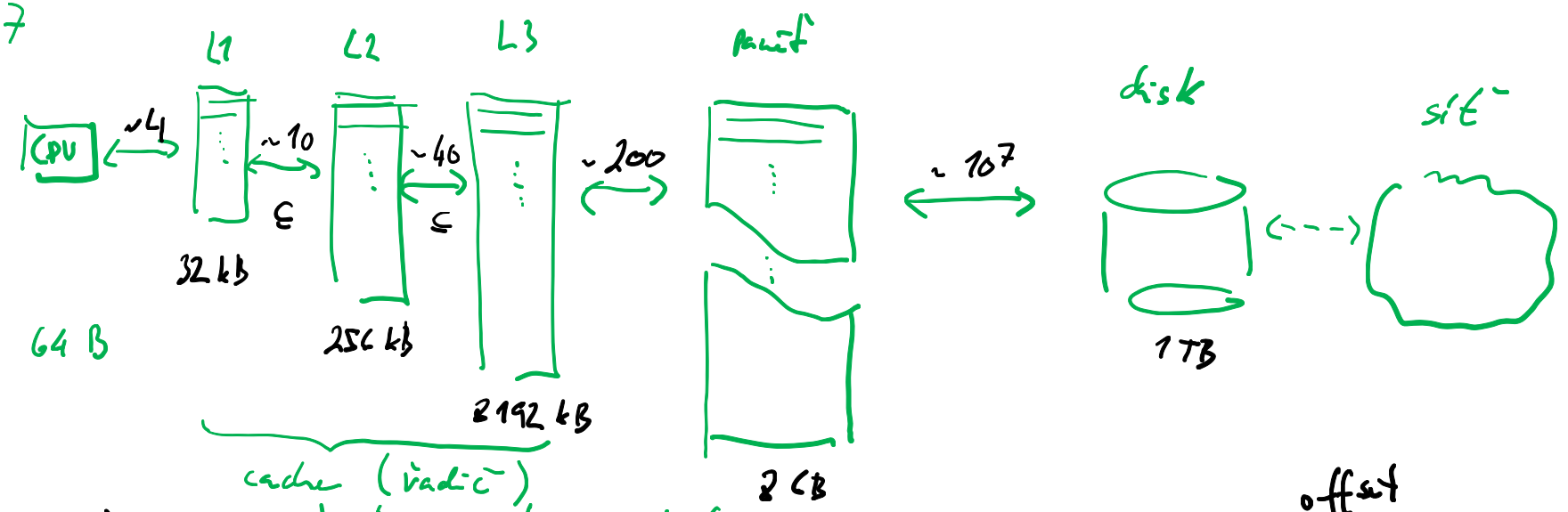
Paměťová hierarchie

Př: Intel Core i7



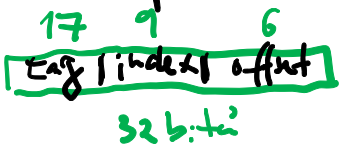
velikost cache ~ 64 B (blok)

mapování (aliasing):

1) plně asociativní – každý blok může být v každé řádce

(4 GB adresovatelný prostor)

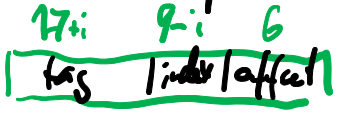
2) přímé mapování – každý blok může být jen v jedné řádce



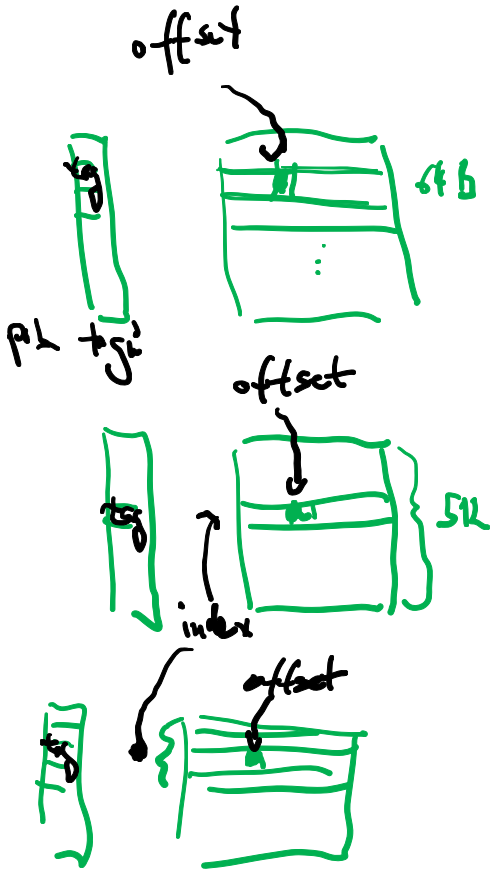
cache 32 kB => 512 řádků

(žádná strategie)

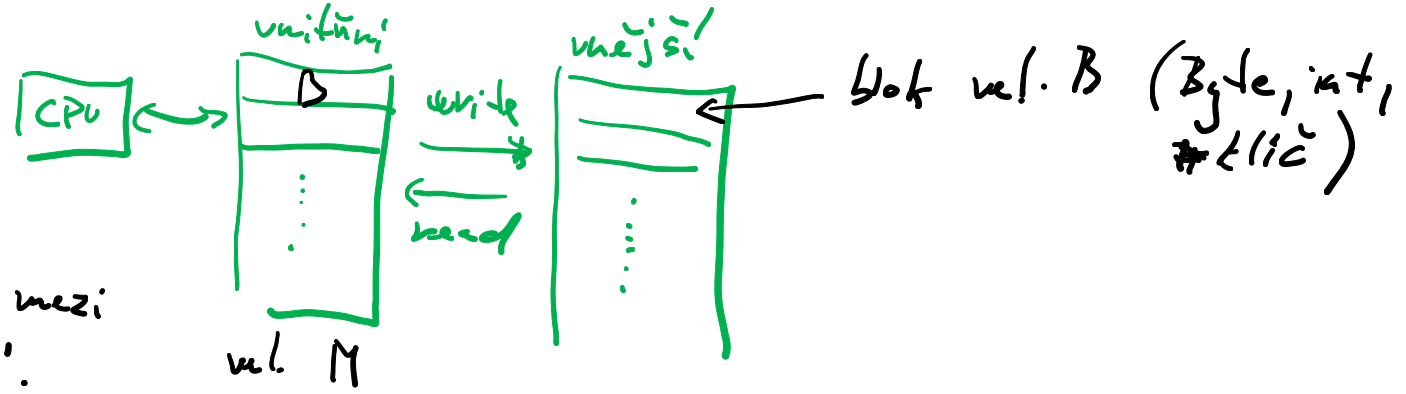
3) množinové mapování – každý blok může být jen v dané množině řádků



(2^i cestná cache)



Modely paměti



1) model čtení paměti (I/O-model)

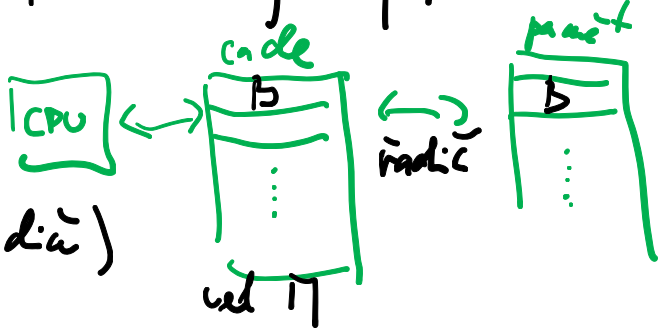
- Alg. (CPU) řídí přenos mezi vnitřní a vnější pamětí.

Def: Alg. má **I/O složitost** $f(n, B, M)$ pokud k parametrům B, M vstupu vel. n vykoná nejvýše $f(n, B, M)$ I/O operací.

Pozn: typicky, #zápisů = $O(\#čtení)$ (pokud $|výstup| = O(|vstup|)$)

2) cache-aware model

- Alg. zná B, M ale řídí přenos mezi cache a pamětí (nezná strategii řadiče)



3) cache-oblivious model

- Alg. nezná B, M (předpoklad $M = \Omega(B^2)$ ... "štíhlá" cache)

$\Rightarrow$ větší nežli, dobrý na hierarchiích cache

Předpoklady (pro cache modely)

1) řadič má optimální strategii (offline): → zná dopředu posloupnost čtení a zápisů bloků ← neznáme (často)

„vyhoď řádek, který bude třeba nejpozději"

2) plně asociativní cache

Průchod pole (příklad)

1) I/O model: zarovnání; stačí 1 řádka (blok)

$\Rightarrow$ I/O-složitost $= \lceil N/B \rceil \le N/B + 1$

2) cache-aware: zarovnání (známe B)

$\Rightarrow$ I/O-složitost = # čtení OPT strategie $\le$ # čtení při strategii $\le N/B+1$

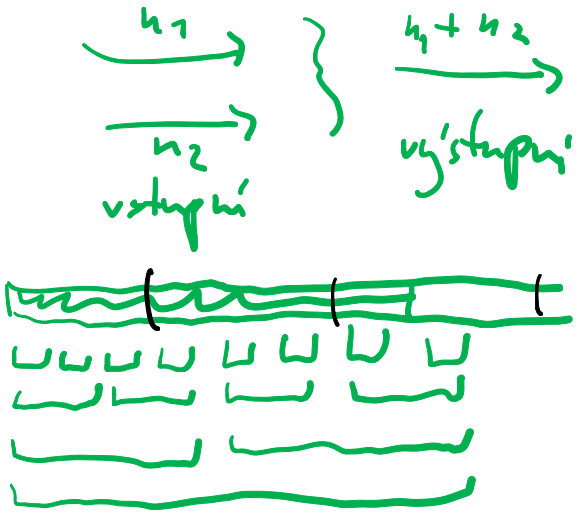
3) cache-oblivious: neznáme zarovnání

$\Rightarrow$ 1 čtení navíc $\le N/B + 2$

$\Rightarrow$ ve všech modelech $O(N/B + 1)$ přenosů

(1 ← lze vynechat)

Merge sort



1 merge: 3x průchod pole
stačí 3 bloky
čas $O(n_1+n_2)$, I/O složitost $O((n_1+n_2)/B+1)$

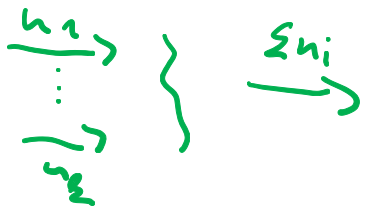
1 fáze: $\Theta(N)$ čas ← pouze pro čtení posledního bloku
$O(N/B+1)$ I/O složitost

→ Merge sort v čase $O(N \log N)$ a I/O složitost $O(N/B \log N + \cancel{0} N)$
1

pro $N \geq B$ ... $O(N/B+1) = O(N/B)$
$N < B$ ... celé pole v 1 bloku
} stačí

Jak využít "větší" část cache?

k-cestný merge sort



potřebujeme $\lceil \log_k N \rceil$ fází = $\lceil \log N / \log k \rceil$

minimum z k prvků (halda) = $\Theta(\log k)$ cas

1 fáze ... v čase $O(N \log k)$

celkem ... v čase $O(N \log k \cdot \log N / \log k) = O(N \log N)$

I/O složitost: (dostatečně velká cache)

1 merge: $O(\Sigma n_i / B + 1)$ přenosů

1 průchod: $O(N/B + 1)$ přenosů — po sobě jdoucí bloky jsou ukládány za sebou

celkem: $O(N/B \log N/\log k + 1)$ pro $N < B$ stačí $O(1)$

k+1 bloků pro skládání + k-1 bloků pro haldu $\Rightarrow$ $M \geq 2Bk$

1) I/O model + cache-aware: zvol $K = \lfloor M/2B \rfloor$

$\Rightarrow \Theta\left(N/B \frac{\log N}{\log M/B} + 1\right)$ I/O složitost (optimální)

2) v cache-oblivious modelu nelze (Frigo et al. '99)