Pamięć w systemach wieloprocesorowych z pamięcią współdzieloną

Rafał Walkowiak

Wersja: 2019/2020

Ograniczenia efektywności systemu pamięci

- Parametry pamięci :
  - opóźnienie (ang. latency) - czas odpowiedzi pamięci na żądanie danych przez procesor
  - przepustowość systemu pamięci (ang. bandwidth) - ilość danych dostarczana przez pamięć w jednostce czasu
  - wielkość linii pamięci podręcznej (ang. cache line) - liczba sąsiednich słów **pobierana jednorazowo do pamięci podręcznej** z pamięci głównej w sytuacji, gdy w pamięci podręcznej procesora nie ma danych z interesującego procesor zakresu adresowego (realizacja kodu i brak trafienia).

Opóźnienie dostępu do pamięci powoduje często spadek efektywności przetwarzania systemu w zależności od rodzaju kodu; w szczególności od:

- wymaganej w kodzie liczby dostępów do pamięci przypadającej średnio na instrukcję (głównie odczytów) (parametr to wskaźnik dostępu - memory access rate) i
- miejsca położenia danych (z którego dane muszą być dostarczone do L1pp).

Ograniczenia efektywności przetwarzania - system pamięci

Pamięć podręczna (pp):

- Zmniejszenie wypadkowego opóźnienia dostępu do pamięci poprzez zastosowanie pamięci podręcznej procesora (ppp). Procesor realizuje dostęp do systemu pamięci komputera (odczyt i zapis) poprzez pamięć podręczną procesora (pierwszego poziomu) - pp L1
- Stosunek trafień do pp (ang. hit ratio) – określa iloraz;
  - liczby odwołań do pamięci w sytuacji: gdy żądana linia była już w pamięci podręcznej i
  - liczby wszystkich odwołań do pamięci.

Wzór na średni czas dostępu do słowa w pamięci - T

- h – stosunek trafień (ang. hit ratio)
- t_{pp} - czas dostępu do pamięci podręcznej
- t_m - czas dostępu do pamięci głównej (czas ten pozwala na pobranie danej do pamieci podrecznej procesora L1)
- $T = h\ t_{pp} + (1-h)\ t_m$
- h=0? h=1? Przykład kodu ?

Przykład analizy jakości dostępu do pamięci

```
Float a[1000];
for (i=0; i<n; i++)
Suma+=a[i]];
```

- Suma lokalna – w rejestrze
- Zmienne Suma, i, n bez dostępu do pamięci
- Tablica w pamięci – kolejne potrzebne wartości ulokowane na sąsiednich adresach pamięci
- Linia pp procesora - 64 Bajty
- Rozmiar typu float - 4 bajty
- Pobrania linii pamięci podręcznej (64B) z elementami tablicy a[] do L1
- Średnio 15 trafień do pamięci podręcznej na 16 kolejnych odczytów z pamięci
- Stosunek trafień do pamięci podręcznej 15/16

Przykład analizy jakości dostępu do pamięci

```
Float a[1000];
for (i=0; i<n; i+=16)
Suma+=a[i]];
```

- Linia pp 64 Bajty
- Rozmiar typu Float 4 baty
- Pobrania linii pamięci (64B) z elementami tablicy a[] do L1
- Używane elementy oddalone są od siebie o 16*4= 64 B
- podczas odczytu tablicy **a** występuje 0 trafień do pp
- Dla tego kodu stosunek trafień do pp = 0

Ograniczenia efektywności przetwarzania - system pamięci

- Pozytywny efekt zastosowania pp procesora wynika z:
  - wielokrotnego wykorzystania danych z pp (szybki dostęp) sprowadzonych jednokrotnie - jako linia pp (wolny dostęp)

Cechami programów, które to zapewniają są:

- czasowa lokalność odwołań (ang. temporal locality of reference) – dane raz sprowadzone do pamięci zostaną użyte wielokrotnie zanim zostaną z pamięci usunięte lub unieważnione, brak ***clo*** powoduje niski stosunek trafień do pp i spowalnia przetwarzanie.
  - przetwarzanie powinno być podzielone na etapy, w których wykorzystywane są dane (rozmiar) mogące być obsłużone przez efektywnie dostępne pamięci i bufor translacji adresów
- przestrzenna lokalnością odwołań - (ang. spatial locality of memory access) korzystanie w kodzie z danych zajmujących sąsiednie lokacje w pamięci – brak ***plo*** powoduje niski stosunek trafień do pp i niski stosunek trafień do bufora translacji adresów.
  - jeżeli tablica jest zapisywana wierszami w pamięci to kolejne dostępy do tablicy powinny też, jeśli to możliwe, być realizowane wierszami,

Współdzielenie danych w systemach równoległych

Powielenie danych w pamięciach podręcznych

- W systemach równoległych, w których procesory współdzielą pamięć dane współdzielone mogą być powielone w wielu pamięciach podręcznych procesorów - te same linie pp w różnych pamięciach.
- Zalety replikacji:
  - Obniżenie opóźnienia dostępu i wymagań przepustowości systemu pamięci operacyjnej
  - Mniej rywalizacji o dane odczytywane przez wiele procesorów – możliwy jednoczesny odczyt wartości przez procesor z pamięci (każdy procesor ze swojej pp)

Współdzielenie danych w systemach równoległych

Strategie zapisu danych przez procesora w pamięci komputera

- Write-through cache – zapis do pp jest synchronicznie odzwierciedlany w pamięci globalnej - każdy zapis uaktualnia pamięć operacyjną (rzadko spotykany)
- Write-back cache – zapis w pamięci operacyjnej jest dokonywany w wyniku zewnętrznego (generowanego przez inny procesor) żądania odczytu przez inny procesor zapisanej linii danych
- Spójność systemu pamięci jest zapewniona dzięki temu, że:

- dane zapisane przez jeden procesor są udostępniane przez system pozostałym procesorom do momentu ponownego zapisu,
- wszystkie procesory widzą taką samą kolejność realizacji przez inne procesory zapisów i mają zapewniony odczyt aktualnej wartości (pewne zapisy mogą być niewidoczne dla procesorów – jeśli zmienionych danych nie żądają).

Protokoły zapewnienia spójności pamięci w systemach wieloprocesorowych

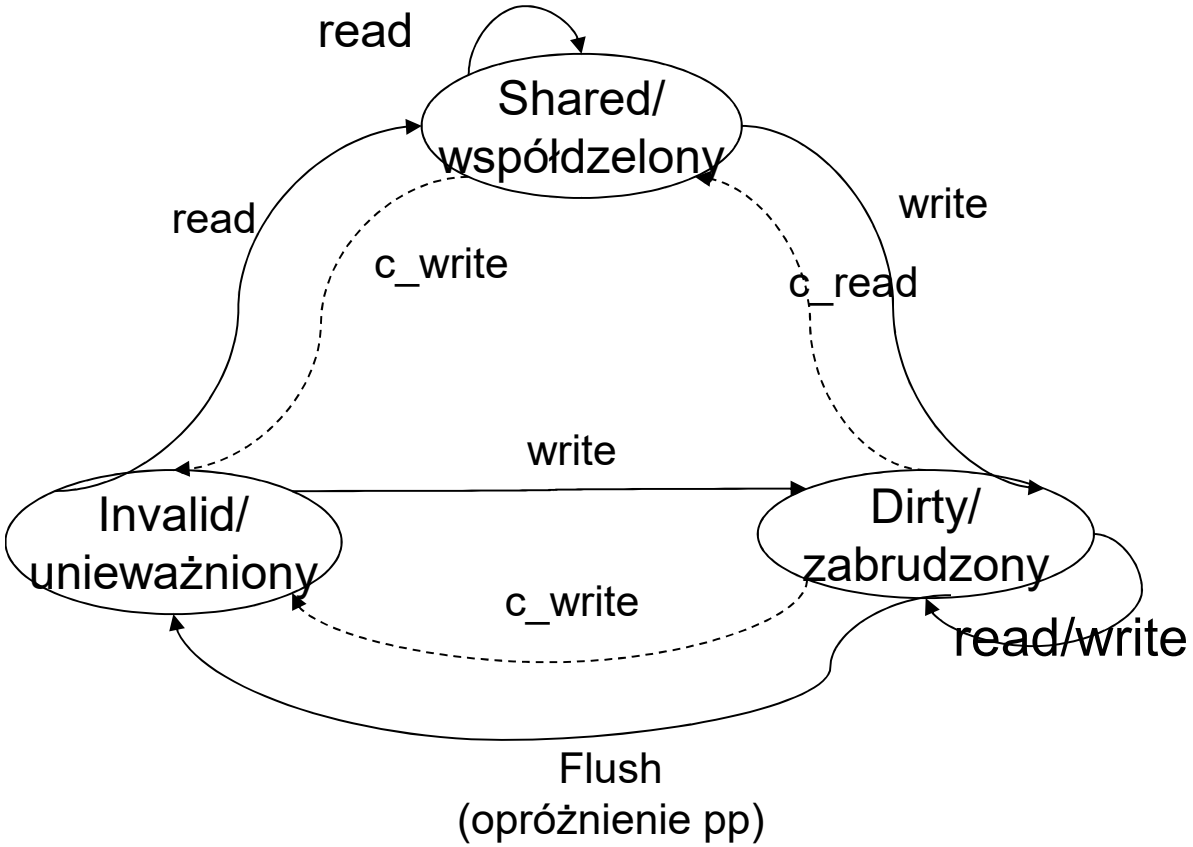
- Protokoły: unieważniania lub uaktualniania służą dla zapewnienia spójności danych współdzielonych - zapewniają istnienie szeregowego porządku wykonania instrukcji realizowanych współbieżnie.
- *Protokół unieważniania*

Powoduje w przypadku zapisu lokalnej kopii danych (linia pp) unieważnienie pozostałeych kopii danych – konsekwencja to wstrzymywanie przetwarzania ze względu na oczekiwanie na dane unieważnione przy żadaniu dostępu do tych danych. Częściej stosowany obecnie ze względu na duże znaczenie przepustowości pamięci i magistrali (ten protokół obniża wymagania na przepustowość, może obniżać prędkość przetwarzania wątku).

- *Protokół uaktualniania*

Powoduje w przypadku zapisu lokalnej kopii danych (linia pp) uaktualnienie pozostałych kopii danych - konsekwencja to narzut komunikacyjny wynikający z przesyłania danych, które nie zawsze będą wykorzystywane; przesłania nowej zawartosci linii pp następują przy każdej modyfikacji każdego słowa wielosłowowej linii pp (efektem jest więcej przesłań i wzrost wymagań na przepustowość systemu pamięci).

3 stanowy protokół zapewnienia spójności danych powielonych w pamięci systemu wieloprocesorowego



Linia pamięci podręcznej przechodzi między stanami na skutek instrukcji realizowanych przez lokalny procesor (read/write/flush) oraz akcji protokołu zapewnienia spójności w odpowiedzi na działania innych procesorów.

W wyniku zapisu słowa linia pp jest lokalnie oznaczona jako Dirty w celu zapewnienia, że procesor ten obsłuży kolejne żądania dostępu innych procesorów do danych w tej linii pp.

Zapis wartości do linii unieważnionej jest poprzedzony pobraniem do pp aktualnej zawartości linii, w której zmienna się znajduje. Procesor posiadający wersję aktualną linii pp udostępnia ją żądającemu procesorowi i dokonuje zapisu linii do pamięci operacyjnej (w przypadku: write back cache).

Protokół zapewnienia spójności danych - przykład

Procesor1	Processor2	PP Proc1	PP Proc2	PAMIĘĆ
				x= 5, D y=12,D
read x	read y			
write x=x+1	write y=y+1			
read y	read x			
write x=x+y				
	write y=x+y			

Kolejne (czas) kroki przetwarzania – kolejne wiersze tabeli
Read – odczyt z pamięci do rejestru
Write – zapis wartości wyznaczonej przez procesor do pamięci
Procesory mają prywatne pamięci podręczne. Zmienne należą do różnych linii pamięci podręcznej, zapisy wymagają uzyskania przez procesor dostępu w trybie wyłącznym do zapisywanej linii
D - Dirty – linia „zabrudzona” - obszar zmodyfikowany - wyłączny dostęp
S -Shared – linia współdzielona - obszar „współdzielony„
I - Invalid – linia nieważna – obszar „nieważny”

Protokół zapewnienia spójności- przykład

Proces1 ~~Procesor1~~	Proces2 ~~Processor2~~	PP Procesor 1	PP Procesor2	PAMIĘĆ
				x= 5, D y=12,D
read x	read y	x=5,S	y=12,S	x= 5, S y=12,S
write x=x+1	write y=y+1	x=6,D	y=13,D	x= 5, I y=12,I
read y	read x	x=6,S y=13,S	x= 6,S y=13,S	x= 6, S y=13,S
write x=x+y		x= 19,D y=13,S	x= 6,I y=13,S	x= 6, I y=13,S
	write y=x+y	x= 19,D y=13,I	x= 6, I y=19,D	x= 6, I y=13,I

Read oznacza odczyt wartości z pamięci. Write oznacza zapis do pamięci wartości wyznaczonej w oparciu o dane lokalnie dostępne w rejestrze.
Protokół zapewnienia spójności zapewnia tyko jedną uznawaną jako poprawna wartość zmiennej w pamięci, mimo że są 3 miejsca jej przechowywania. Zmienna przechowywana w rejestrze może mieć równocześnie inną wartość niż zmienna w pamięci. Wyścig w dostępie do danych.

Protokół zapewnienia spójności danych - przykład

Proces1 Procesor1	Proces2 Processor2	PP Procesor 1	PP Procesor2	PAMIĘĆ
				x= 5, D y=12,D
read x	read y	x=5,S	y=12,S	x= 5, S y=12,S
write x=x+1	write y=y+1	x=6,D	y=13,D	x= 5, I y=12,I
read y	read x	x=6,S y=13,S	x= 6,S y=13,S	x= 6, S y=13,S
write x=x+y		x= 19,D y=13,S	x= 6,I y=13,S	x= 6, I y=13,S
	write y=x+y	x= 19,D y=13,I	x= 6, I y=19,D	x= 6, I y=13,I

Przykład powyższy kodu można sklasyfikować jako charakteryzujący się wyścigiem w dostępie do danych, gdyż nie ma synchronizacji wątków w dostępie do zapisywanych i odczytywanych wartości tej samej zmiennej (x,y) przez różne wątki. Możliwy ze względu na brak synchronizacji i faktycznie widoczny w powyższym scenariuszu przetwarzania jest fakt użycia przez proces 2 w ostatnim kroku nieaktualnej wartości zmiennej x=6, która została zapisana chwilę wcześniej przez procesor 1 i ma wartość 19.

Tablica przejść linii pp między stanami w wyniku operacji procesora na danych linii pp

stan	read	write	flush	c-read	c-write
shared	shared	dirty	invalid	shared	invalid
dirty	dirty	dirty	invalid	shared	invalid
invalid	shared	dirty	invalid	invalid	invalid

c-read i c-write ocznaczają efekt działań protokołu zapewnienia spójności, operacja flush powoduje zapisanie linii pp do pamięci operacyjnej i powoduje w przypadku konieczności ponownego wykorzystania linii pp ponowne jej wczytanie.

Implementacja spójności pamięci podręcznej

protokół podglądania (ang. Snoopy cache coherence protocol) .

- Procesor monitoruje przesłania na magistrali dotyczące swoich linii pp.
- Procesor zapisuje lokalnie stan swoich danych.
- Wykrycie przez procesor zewnętrznego żądania odczytu linii, której stan jest lokalnie „dirty” powoduje przesłanie przez procesor lokalnej kopii linii do procesora żądającego odczytu linii pp.
- Jeżeli natomiast w innym procesorze nastąpił zapis do linii pamięci, której kopia jest przechowywana w lokalnej linii pp to następuje unieważnienie lokalne nieaktualnej kopii linii.
- Operacje na linii „dirty” są realizowane lokalnie.
- Mechanizm wymaga rozgłaszania do procesorów informacji o operacjach na pamięci (funkcje zapewnienia spójności). Rozgłaszanie to oznacza w praktyce wzrost kosztu realizacji mechanizmu ze wzrostem liczby procesorów (brak skalowalności) - liczba operacji wzrasta liniowo z liczbą procesorów.

Implementacja spójności pamięci podręcznej

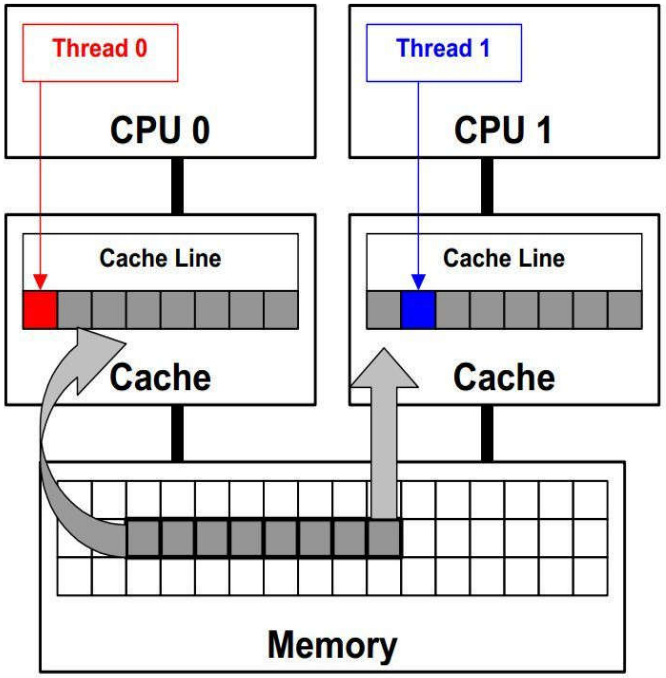
mechanizmy katalogowe

Lepsze rozwiązanie dla mechanizmu zapewnienia spójności

- Lepsza efektywność mechanizmu osiągana w przypadku zapamiętywania *istnienia/braku stanu współdzielenia danych* (czy linia jest współdzielona czy jest wyłączna (dirty) ?)
- Pozwala na obniżenie wymagań przepustowości magistrali – występuje komunikacja tylko do zainteresowanych procesorów.
- Pamięć centralna jest rozszerzona o pamięć katalogową (PK), w której zapisywane są informacje na temat procesorów korzystających z poszczególnych stron pamięci. Te procesory będą uczestniczyły w dystrybucji informacji zapewniających spójność (informacji o odczytach linii pp unieważnionych i zapisach linii pamięci współdzielonych na tych stronach)
- Zapewnienie spójności bazuje na katalogu scentralizowanym lub rozproszonym.
- W przypadku rozproszonych pamięci katalogowych (PK) znika wąskie gardło jakim jest obsługa protokołu spójności w oparciu o jedną PK – wtedy możliwa jest jednoczesna realizacja wielu operacji zapewnienia spójności.

Nieprawdziwe współdzielenie – ang.false sharing

- Nieprawdziwe współdzielenie– sytuacja powodująca dodatkowe narzuty czasowe wynikajace z unieważnień kopii danych poprzez zapisy przez różne procesory różnych słów ulokowanych logicznie w tym samym obszarze linii pp – efektem zapisu jest unieważnienie wszystkich nieaktualnych kopii zapisanej linii (w innych procesorach).
- W procesorach, gdzie linie zostały unieważnione, gdy wystąpi żądanie dostępu (odczyt lub zapis) do danych zawartych w unieważnionych liniach następuje wstrzymanie dostępu do tych danych do momentu sprowadzenia do pp linii w postaci ostatnio zmodyfikowanej.
- Szczególnie kosztowne jest wielokrotne wystąpienie powyższej sytuacji wynikającej z zapisu w różnych procesorach – spadek efektywności przetwarzania



Wielokrotnie realizowane z przeplotem zapisy czerwony i niebieski – wymaganie uaktualnienia zawartości unieważnionej kopii linii pp.

Efektywność dostępu do pamięci – pamięć podręczna danych podsumowanie

Kategorie braków trafień do pamięci podręcznej procesora

- Braki trafień pierwszego dostępu (ang. compulsory misses)– pierwsze odwołanie do jednostki danych, poprawa sytuacji: wyprzedzające pobranie danych (realizowane przez kompilator lub procesor – możliwe poprzez analizę sposobu dostępu w kodzie - on-line lub off-line)
- Braki trafień wynikające z pojemności pamięci podręcznej (nie linii pp) (ang. capacity misses) – poprawa efektywności możliwa poprzez wzrost lokalności kodu:
  - zmniejszenie wykorzystywanej przestrzeni danych („zagęszczenie danych")
  - podział przetwarzania na etapy - przy braku możliwości pomieszczenia w pamięci podręcznej wszystkich danych używanych cyklicznie, określenie etapów przetwarzania ze zbiorami danymi: wielokrotnie używanymi, mieszczącymi się w pamięci podręcznej.
- Braki trafień wynikające z konfliktów (ang. conflict misses) – odwołanie do linii danych po jej unieważnieniu, poprawa - przesunięcie pozycji danych do pozycji nie powodującej konfliktu dostępu – usunięcie false sharing'u lub usunięcie niekorzystnego odstępu (w dostępie do danych) równego wielkości sekcji wielosekcyjnej pp (por. działanie wielosekcyjnej pp).

Zarządzanie pamięcią przez system operacyjny pamięć wirtualna

Pamięć wirtualna

- przydzielana procesom w blokach o wielkości „strony pamięci wirtualnej”,
- udostępniana procesom po zapisaniu danych strony wirtualnej z pliku wymiany do obszaru pamięci operacyjnej -– do tzw. „ramki pamięci”,
- umożliwia przydział procesom większej ilości pamięci niż jest dostępna fizycznie w systemie,
- aby procesor mógł zrealizować dostęp do danych spod adresu wirtualnego konieczne jest odwzorowanie adresu wirtualnego (wynikajacego z kodu) na aktualny adres fizyczny, pod którym dane aktualnie się znajdują w pamięci operacyjnej (lub podręcznej),
- konieczna jest zatem translacja adresów wirtualnych na adresy fizyczne.

Dostęp do pamięci wirtualnej - bufor translacji adresu (TLB)

- Każdy dostęp procesora do pamięci powoduje konieczność określenia fizycznego adresu pod którym znajduje się wartość dla określonego w kodzie adresu wirtualnego.
- Odwzorowanie (translacja) jest realizowane przez bufor translacji adresu TLB (ang. translation lookaside buffer), który zawiera pary adresów dla ostatnio translowanych adresów wirtualnych.
- TLB jest strukturą prywatną rdzenia procesora.
- TLB to pamięć podręczna adresów ramki pamięci dla stron wirtualnych
- TLB może posiadać strukturę wielopoziomową i może być oddzielna dla danych i kodu oraz rdzeni procesora.
- W przypadku braku adresu wirtualnego w TLB układy procesora lub system operacyjny korzystając z katalogu i tablic stron określa brakujący adres i wpisuje go do TBL (znaczy koszt czasowy).

Zarządzanie pamięcią - dostęp do pamięci

Brak wymaganej informacji w TLB jest nazywany brakiem trafienia do TLB. Niski stosunek trafień do TLB jest spowodowany niską przestrzenną lokalnością kodu. Np. w programie (język C) odczyt różnych elementów kolumny tablicy z długimi wierszami.

Gdy znany jest adres fizyczny operandu (z TLB) wtedy można określić:

- czy czytana często spekulatywnie z pp L1 (współbieżność działań) wartość jest poprawna – jeśli trafienie do pp,

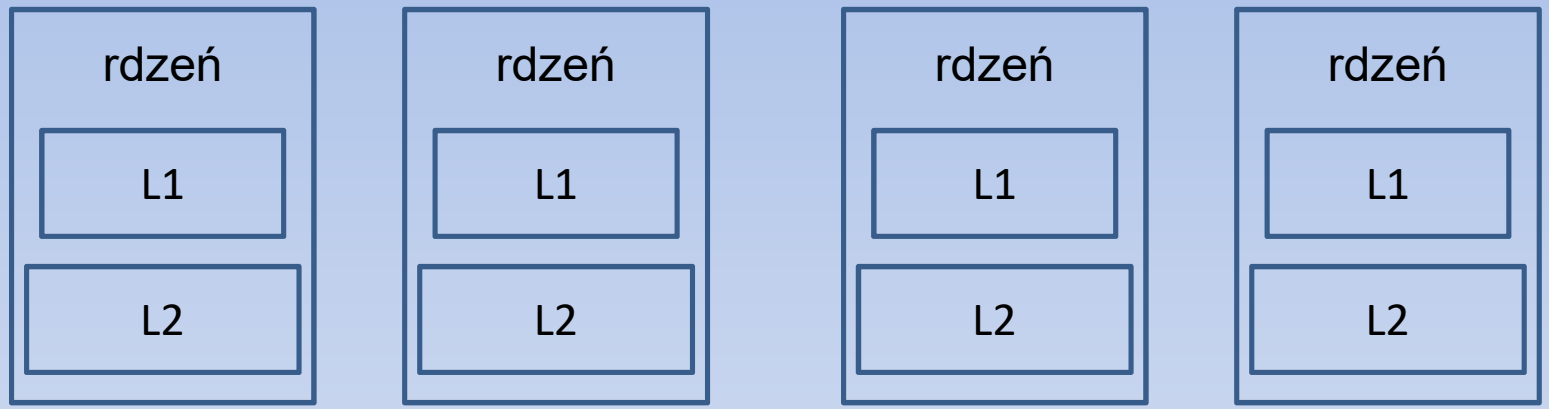
- czy można zrealizować zapis do pp L1 – jeśli trafienie do pp,

- skąd należy pobrać żądaną linię, czy jest w strukturze pp czy trzeba pobrać z pamięci operacyjnej.

W przypadku braku linii z żądanymi danymi w pp L1 brakująca linia (cache miss) jest pobierana z pamięci niższego poziomu pamięci podręcznej (L2, L3) lub pamięci operacyjnej.

W przypadku braku strony z żądanymi danymi w pamięci operacyjnej (page fault) żądana strona odczytywana jest z dysku z pliku wymiany stron.

Poziomy pamięci podręcznej AMD Phenom II X4, Intel Core i5, i7



L3 AMD 6MiB, Intel 8MiB
Używana do wymiany danych między rdzeniami, porównaj false sharing