Continuous Memory Allocator

Alokacja dużych obszarów ciągłej fizycznie pamięci

Michał Nazarewicz

mina86@mina86.com

Instytut Informatyki Politechniki Warszawskiej

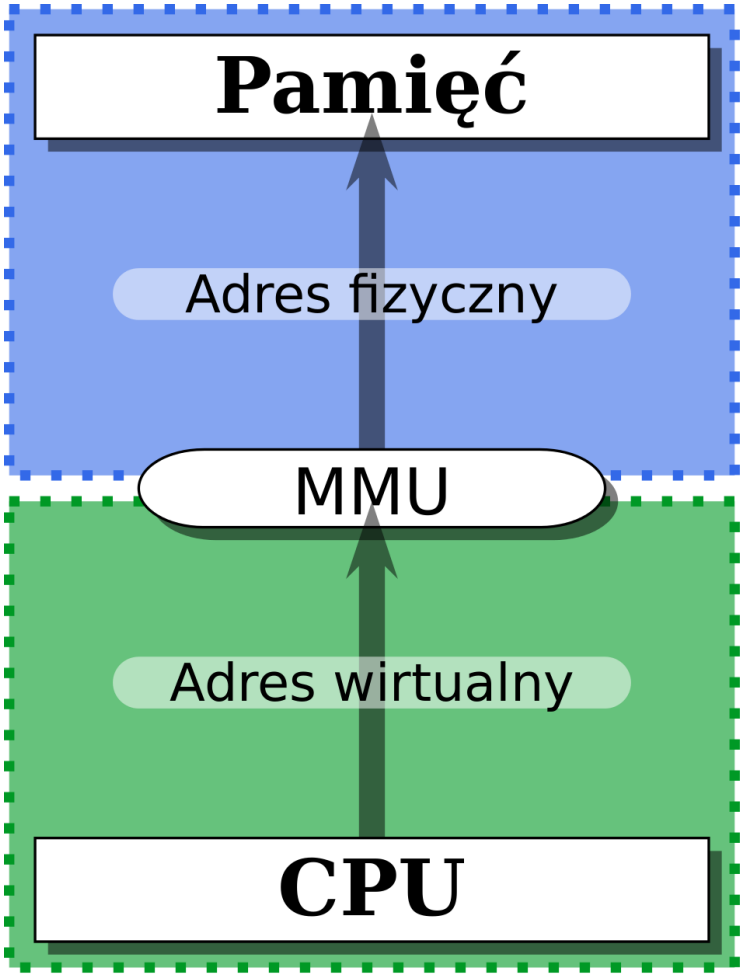
20 marca 2016

Zarys prezentacji

Zarys prezentacji

Wszechmocny MMU

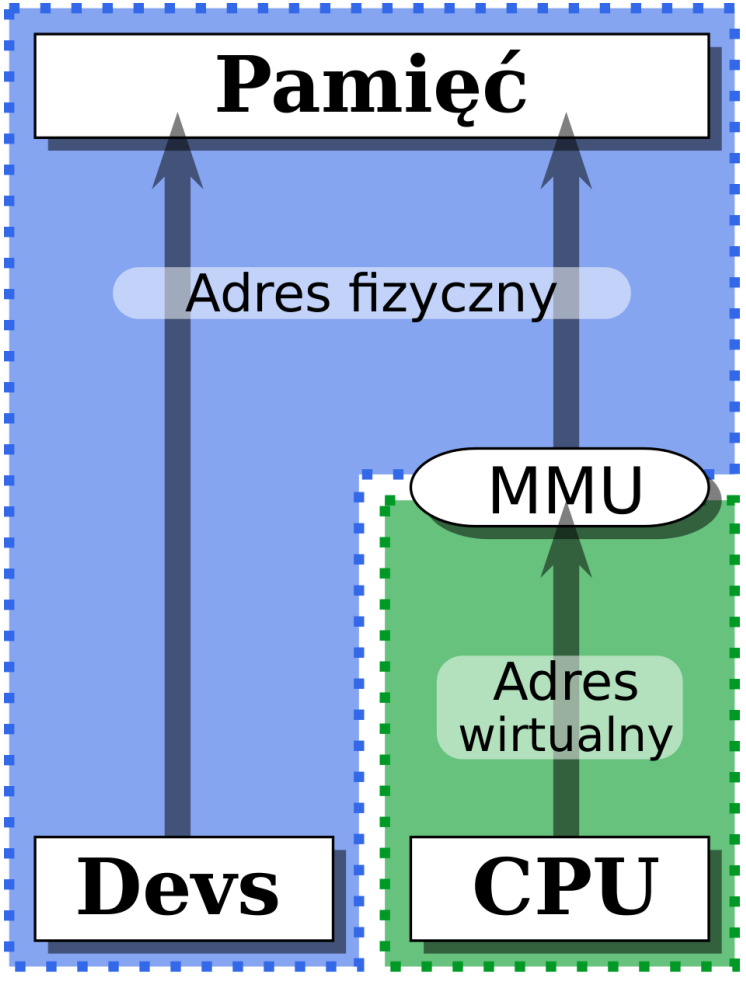
- Współczesne komputery mają MMU.
  - Adres logiczny $\rightarrow$ fizyczny.
- Ciągłe logicznie $\not\Longrightarrow$ ciągłe fizycznie.
- W czym więc problem?



Wszechmocny MMU

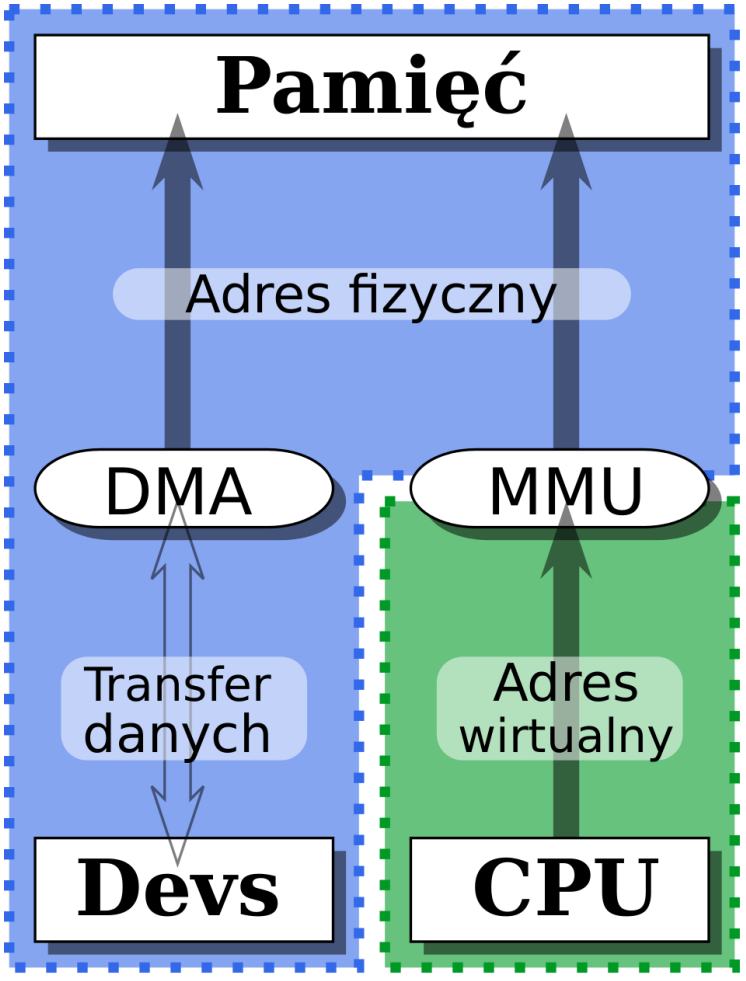
- Współczesne komputery mają MMU.
  - Adres logiczny $\rightarrow$ fizyczny.
- Ciągłe logicznie $\not\Longrightarrow$ ciągłe fizycznie.

- MMU stoi przed procesorem.
- W komputerze mogą być inne komponenty.
- Niektóre wymagają dużych buforów.
  - Np. pięciomegapikselowa kamera.
- W systemach wbudowanych takich komponentów jest sporo.



Wszechmocne DMA

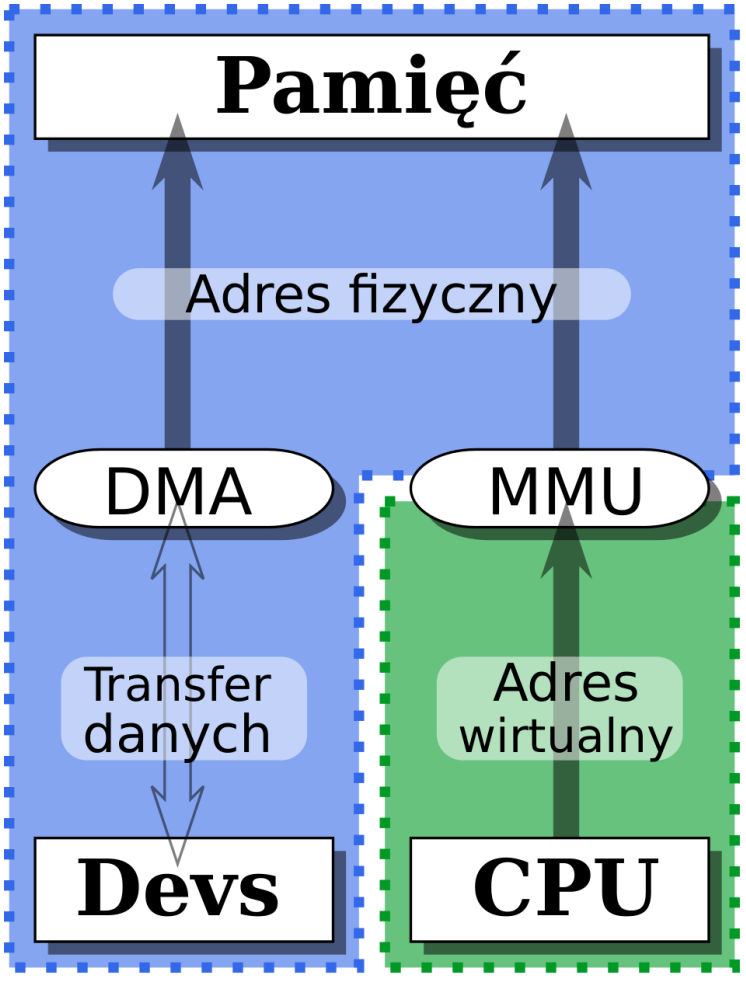
- DMA może udostępniać wektorowe wejście/wyjście.
- Budowanie bufora poprzez zbieranie rozrzuconych części.
- Ciągłe dla urządzenia $\not\Longrightarrow$ ciągłe fizycznie.
- W czym więc problem?



Wszechmocne DMA

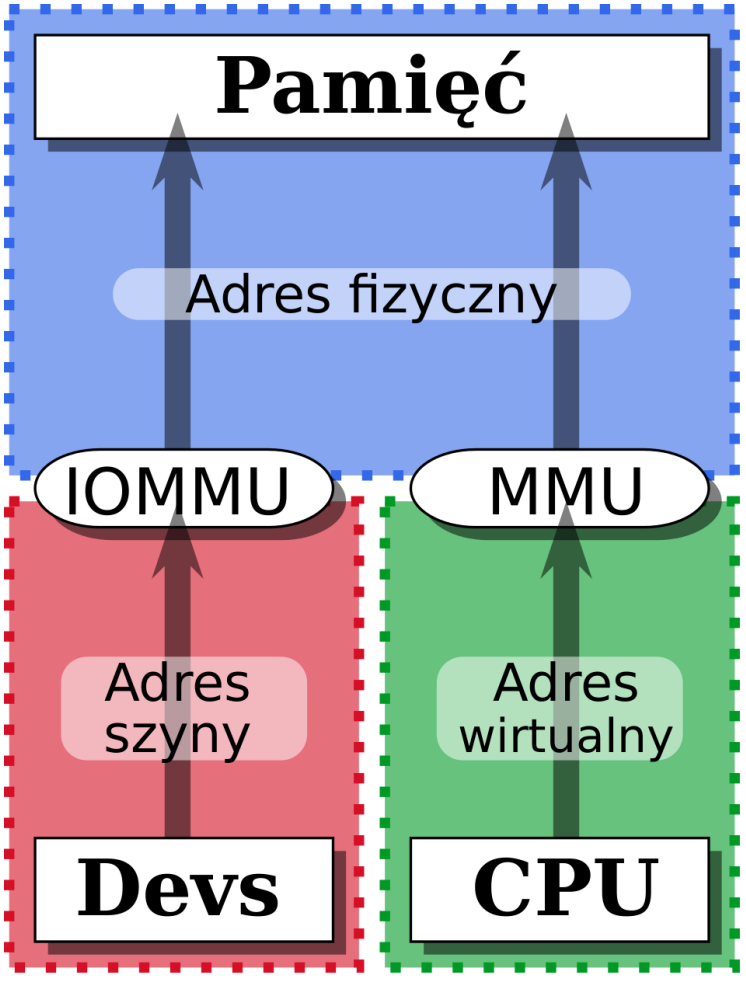
- DMA może udostępniać wektorowe wejście/wyjście.
- Budowanie bufora poprzez zbieranie rozrzuconych części.
- Ciągłe dla urządzenia $\not\Longrightarrow$ ciągłe fizycznie.

- Wektorowe we/wy może być niedostępne.
- DMA działa tylko dla sekwencyjnego dostępu.



Wszechmocny I/O MMU

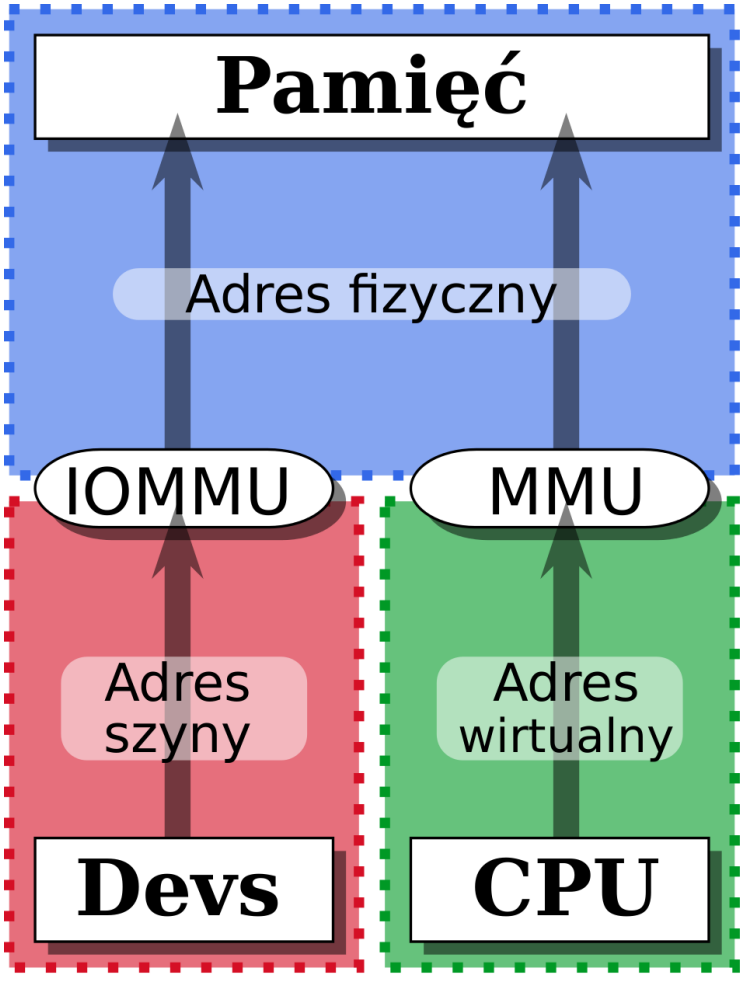
- Czemu nie wstawić MMU przed urządzeniami?
  - Adres urządzenia → fizyczny.
- Działa jak MMU procesora.
- W czym więc problem?



Wszechmocny I/O MMU

- Czemu nie wstawić MMU przed urządzeniami?
  - Adres urządzenia → fizyczny.
- Działa jak MMU procesora.

- I/O MMU nie jest tak powszechne.
- I/O MMU zajmuje czas.
- I/O MMU zużywa energię.



Rezerwacja i przydzielanie przy starcie

- Rezerwacja pamięci przy starcie systemu.
  - Parametr mem.
  - Memblock / bootmem.
- Przypisanie na stałe buforów do urządzeń.
- Gdy urządzenie nic nie robi, pamięć jest marnowana.

Rezerwacja i zarządzanie

- Rezerwacja pamięci przy starcie systemu.
- Zdefiniowanie interfejsu alokatora.
- Mniej pamięci jest zarezerwowanej.
- Niemniej nadal jest marnowana.

- bigphysarea
- Physical Memory Manager

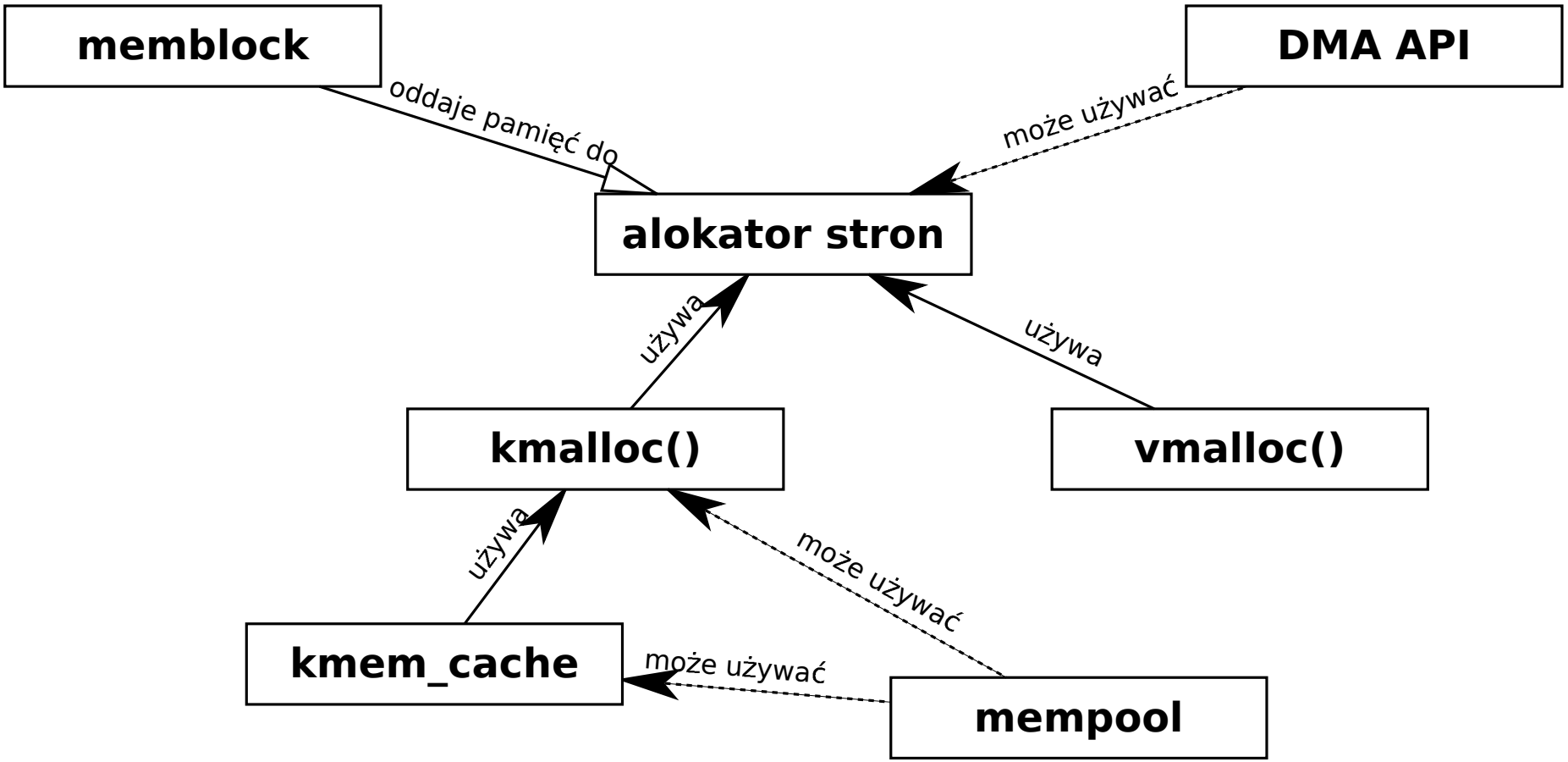
Rezerwacja i zwrócenie systemowi

- Rezerwacja pamięci przy starcie systemu.
- Zwracanie pamięci z powrotem do systemu,
  - ale tylko strony ruchome mogą być z niej alokowane.
- Zdefiniowanie interfejsu alokatora.
- Migruj strony przy alokacji.

- Contiguous Memory Allocator

Zarys prezentacji

Alokatory pamięci Linuksa



kmalloc, vmalloc itp.

- kmalloc
  - Najczęściej stosowany alokator.
  - Ciągłe fizycznie obszary $\leq 4\,\mathrm{MiB}^{\dagger}$.
  - Korzysta z alokatora stron.
- vmalloc
  - Alokacje nieciągłe fizycznie.
  - Korzysta z alokatora stron.
- kmem_cache
  - Alokacja obszarów o z góry zadanym rozmiarze.
  - Inny interfejs dla kmalloc().
- mempool
  - Interfejs do tworzenia podręcznych pul buforów.
  - Zazwyczaj korzysta z kmem_cache i kmalloc().

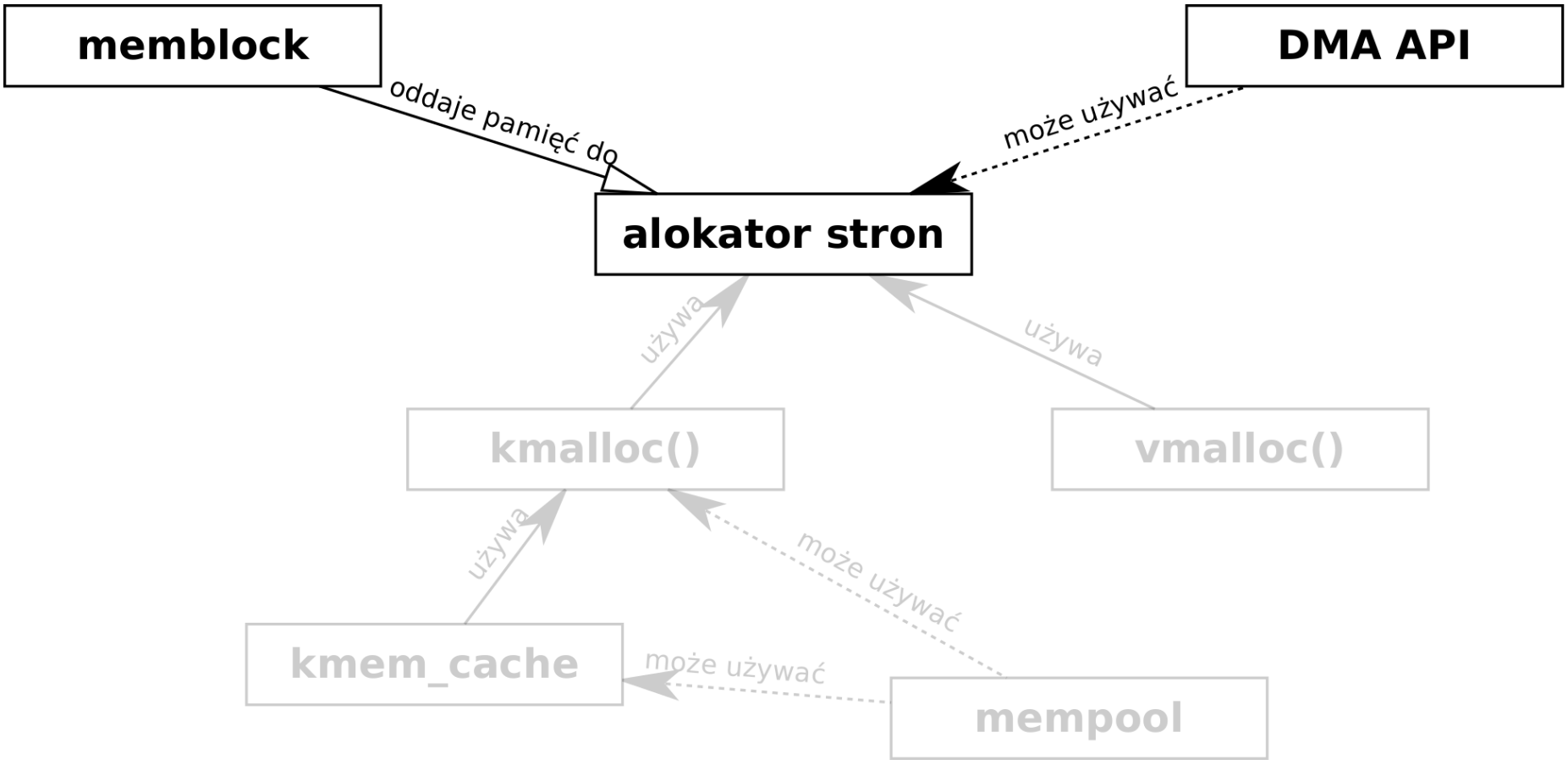
Alokator czasu startu systemu

- Memblock jest aktywny przy starcie systemu.
- Przechowuje listę dostępnych obszarów pamięci RAM.
- Prosty interfejs do alokowania buforów.
- Możliwe jest rezerwowanie dużych bloków.
- Pozostała wolna pamięć jest przekazywana do alokatora stron.

Alokator przeznaczony dla urządzeń

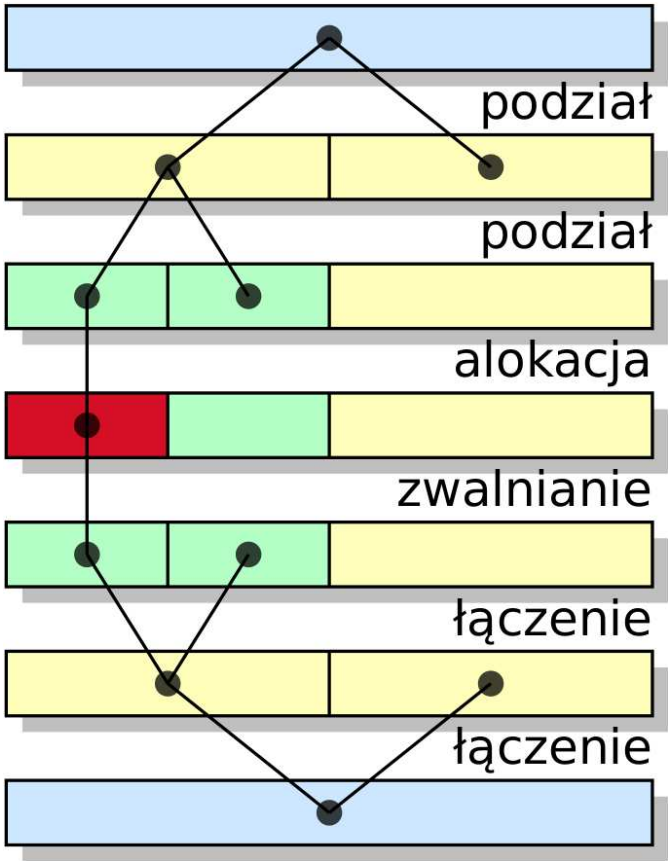
- DMA API umożliwia alokowanie buforów DMA.
- Są one dostępne dla urządzeń na płycie głównej.
  - W szczególności dla kontrolera DMA.
- Implementacja zależna od architektury.
- Często korzysta z alokatora stron.

Alokatory pamięci Linuksa, II

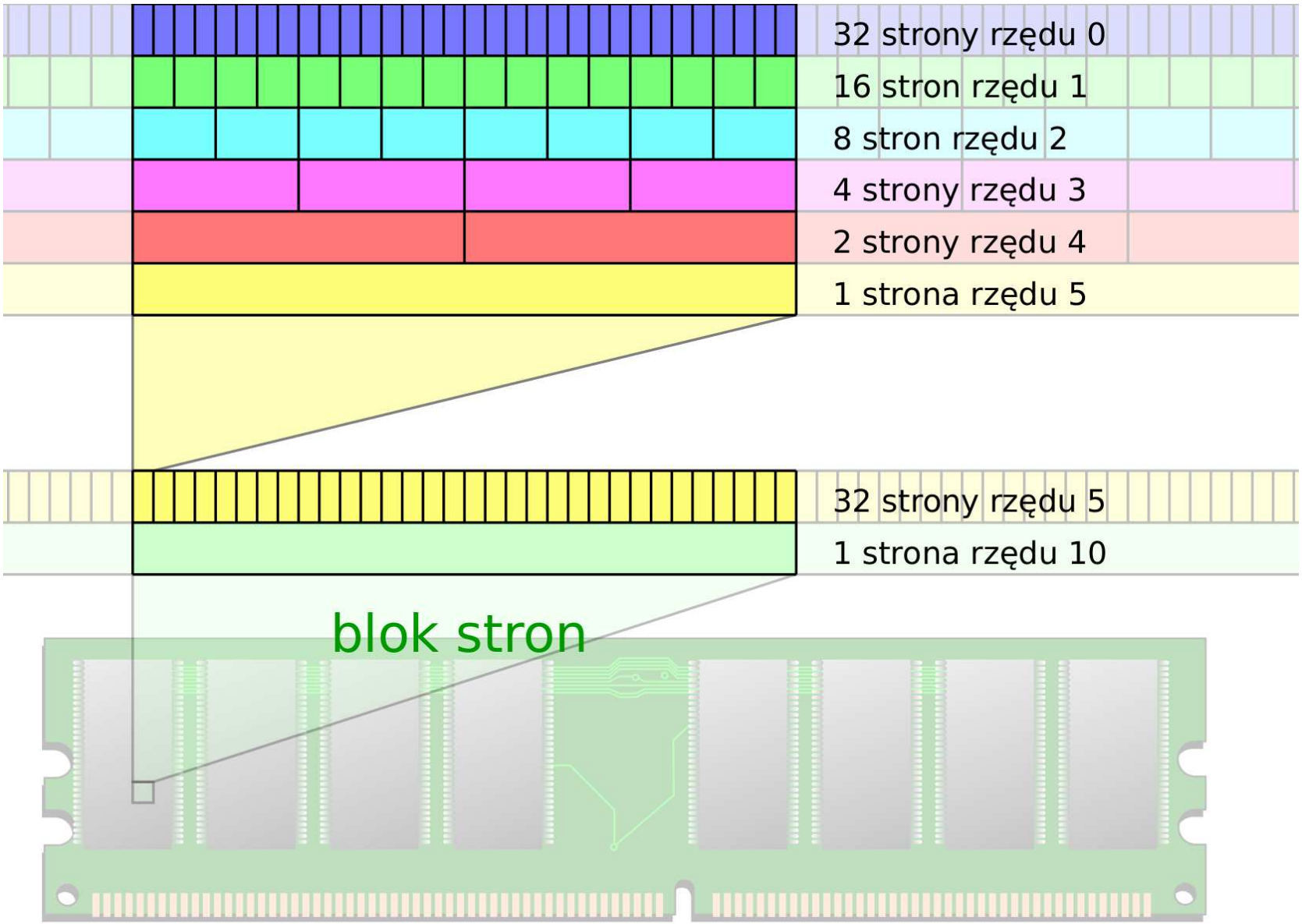


Algorytm bliźniaków

- Alokator stron implementuje algorytm bliźniaków.
- Żądania w kategoriach rzędu strony.
- Rząd może być 0–10[†].
- Zbyt duże strony są dzielone na połowy (tzw. bliźniaków).
- Podczas zwalniania, strona jest łączona z bliźniakami.



Strony i bloki stron



Typy migracji stron

- Przy alokacji, użytkownik określa typ migracji żądanej strony:
  - *unmovable*, *reclaimable* lub *movable*.
- Dla potrzeb CMA:
  - *unmovable* i *reclaimable* $\rightarrow$ strony *nieruchome*,
  - *movable* $\rightarrow$ strony *ruchome*.

Typy migracji bloków stron

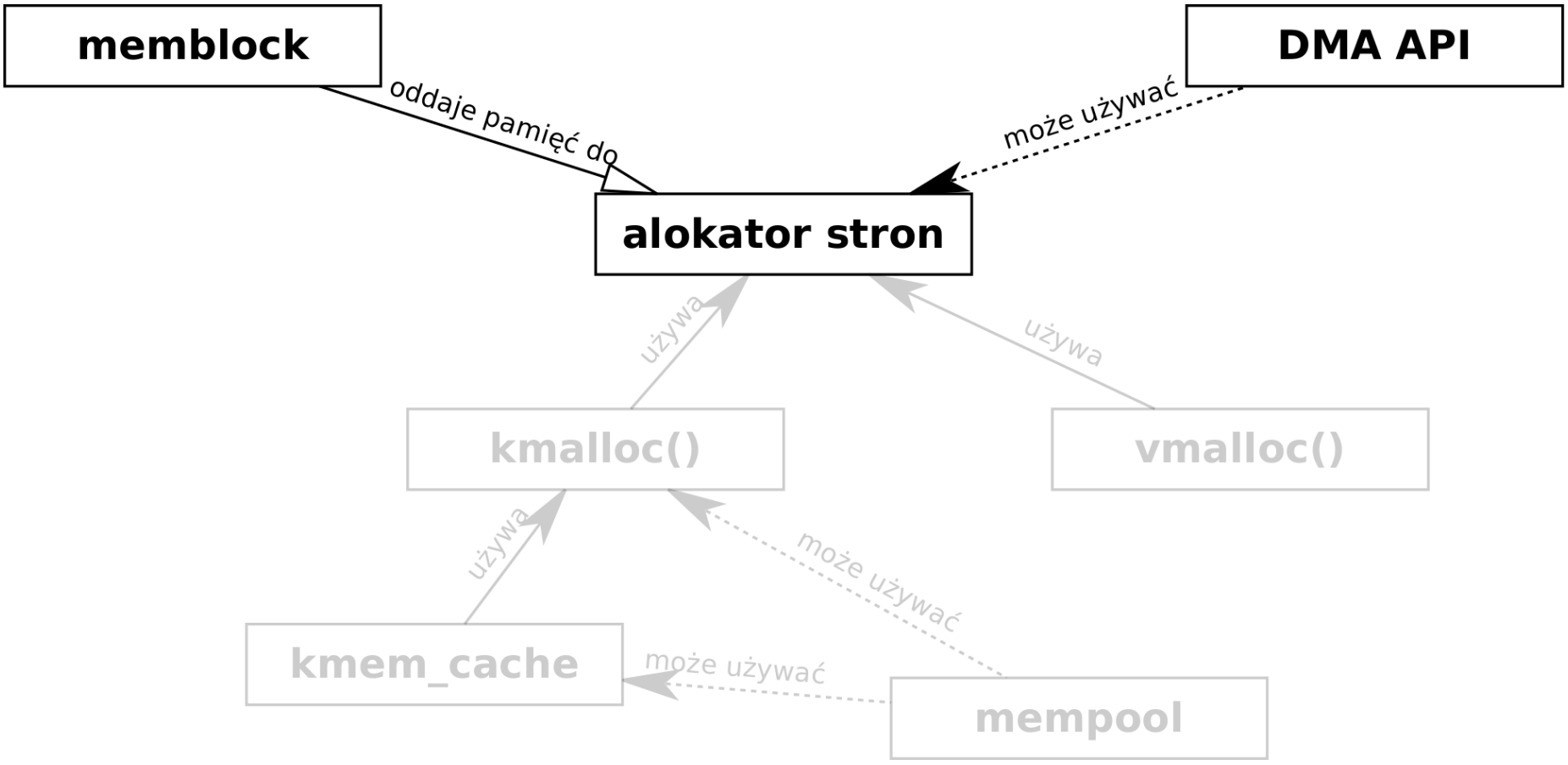
- Każdy blok stron ma przypisany typ migracji.
- Przy alokacji strony są brane z odpowiedniego bloku.
  - Chyba, że takowych nie ma.
- Ułatwia to trzymanie stron tego samego typu razem.
- Typ bloku może się zmieniać.

Migracja

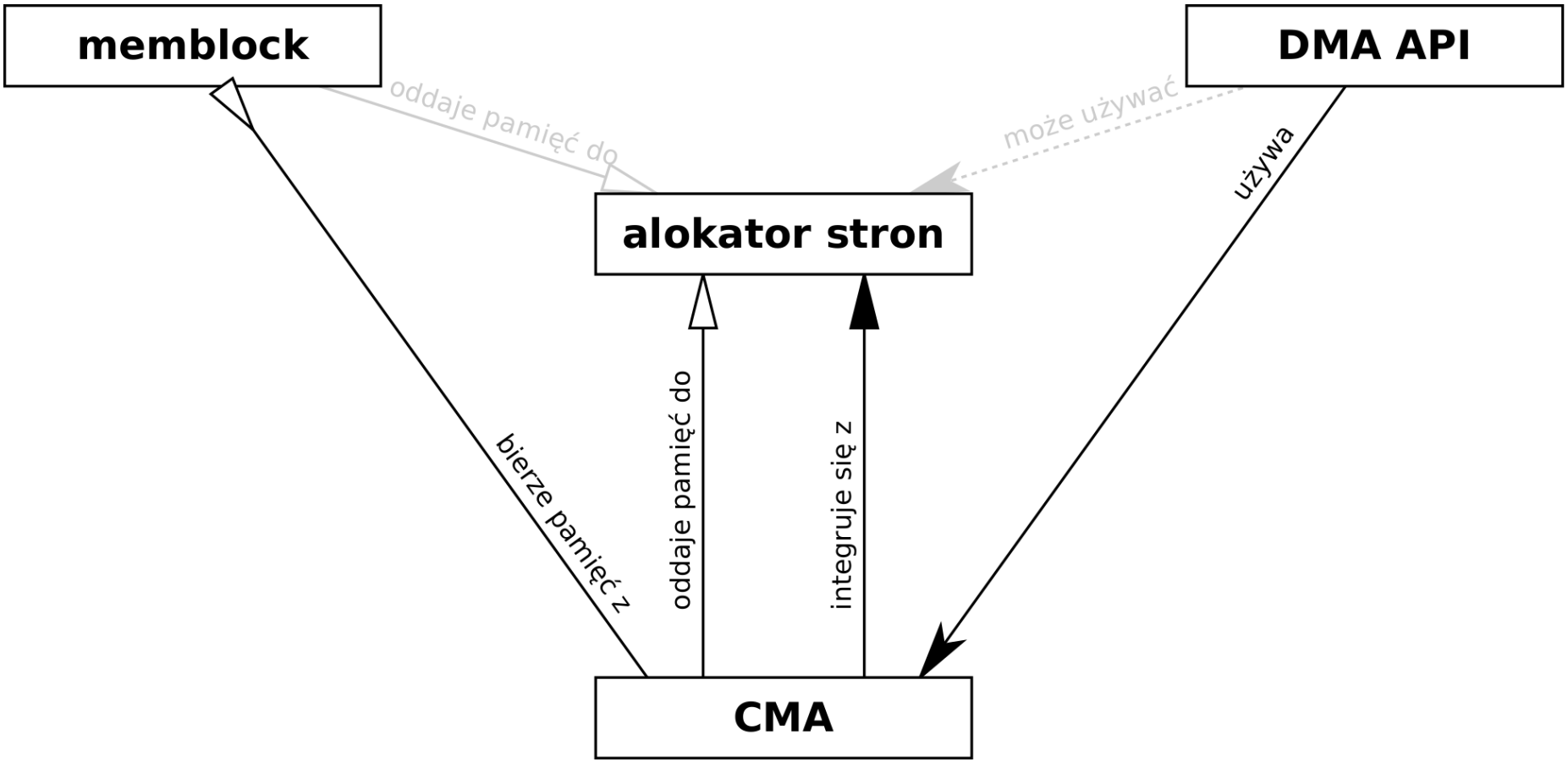
- Strony ruchome mogą być migrowane.
- Migracja polega na:
  - skopiowaniu zawartości strony gdzie indziej i
  - uaktualnieniu odwołań do starej strony.
- Przykładami stron ruchomych są:
  - anonimowe strony procesów i
  - bufory dyskowe.

Zarys prezentacji

Interakcja CMA z innymi alokatorami



Interakcja CMA z innymi alokatorami



Integracja z DMA API

- DMA API umożliwia alokowanie buforów DMA.
- Implementacja zależna od architektury.
- Funkcje DMA API danej architektury muszą być zmodyfikowane by wołać CMA.

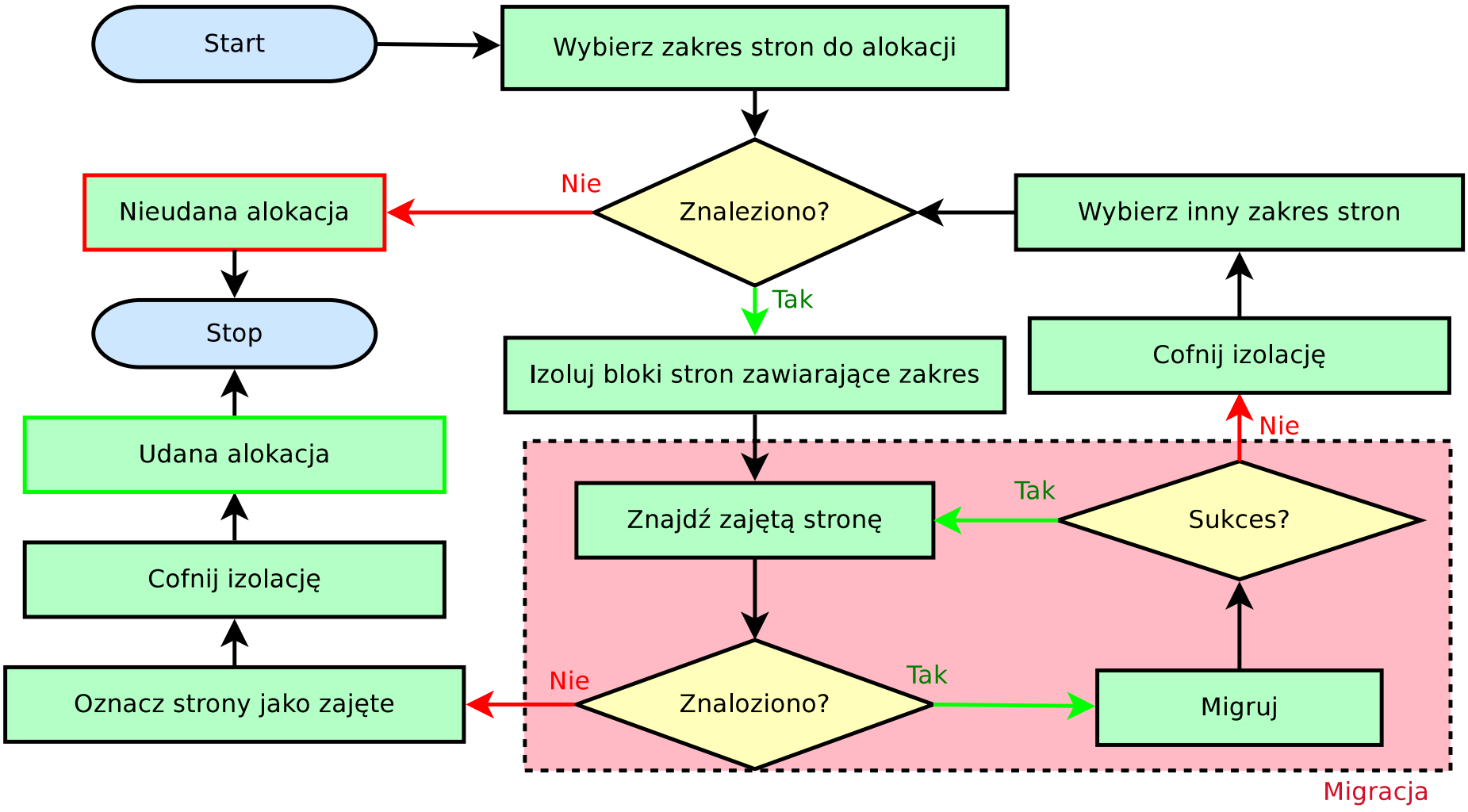
Rezerwacja regionów CMA

- Przy starcie, CMA rezerwuje pamięć z memblock.
  - Zarezerwowany obszar nazywa się regionem lub kontekstem.
- Przygotowany region jest zwracany do puli alokatora stron.
- Kontekst posiada bitmapę stron niezaalokowanych przez CMA.

Typ migracji CMA

- CMA migruje wiele stron na raz.
  - Dostępne typy migracji nie gwarantują ciągłości.
- Nowy typ migracji: *cma*.
- Blok stron tego typu:
  - nigdy nie zmienia typu oraz
  - alokator stron alokuje z niego tylko strony ruchome.

Algorytm alokacji



Problemy

- Strony ruchome nie zawsze można migrować:
  - get_user_pages().
  - Strony dziennika ext4.
  - Niektóre systemy plików.
- Migracja zajmuje czas.

Przyszłość?

- Mądrzejszy dobór zakresu.
- Stronicowanie + zRam.
- Pamięć transcendentna.
- POSIX_FADV_VOLATILE.

Dziękuję!

- Michał Nazarewicz
- mina86@mina86.com, mpn@google.com
- `http://mina86.com/cma/`