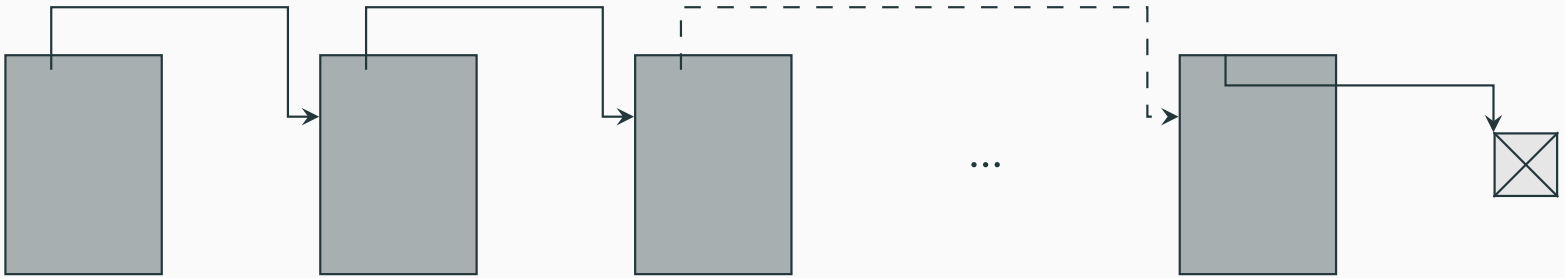
Memory management

CS143A: Principles of operating systems - Fall '17

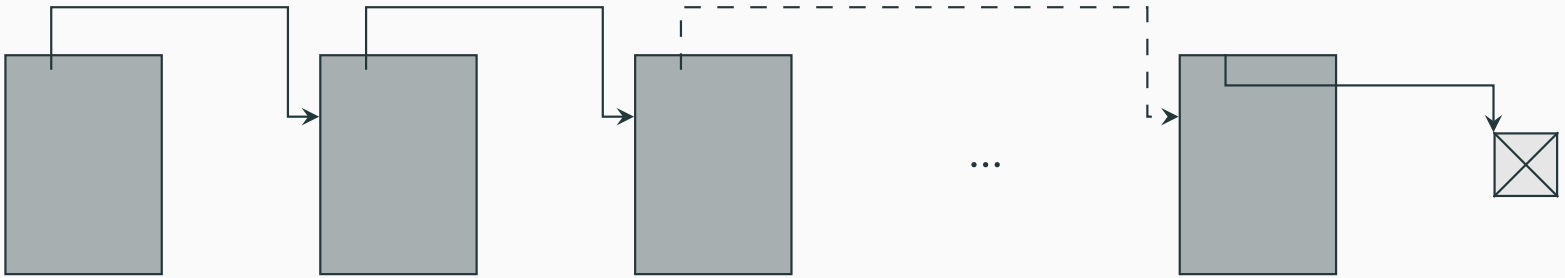
UC Irvine, California

xv6 memory management [1]



[1] kalloc.c

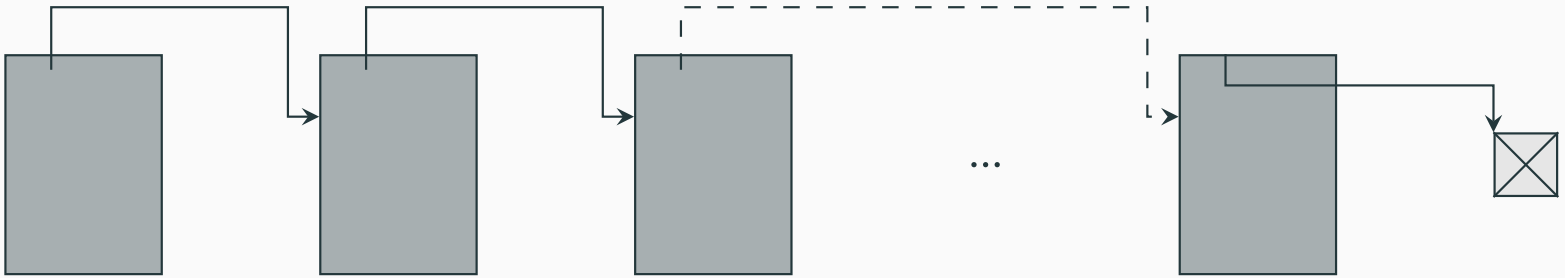
xv6 memory management [1]



- A linked-list of free pages

[1] kalloc.c

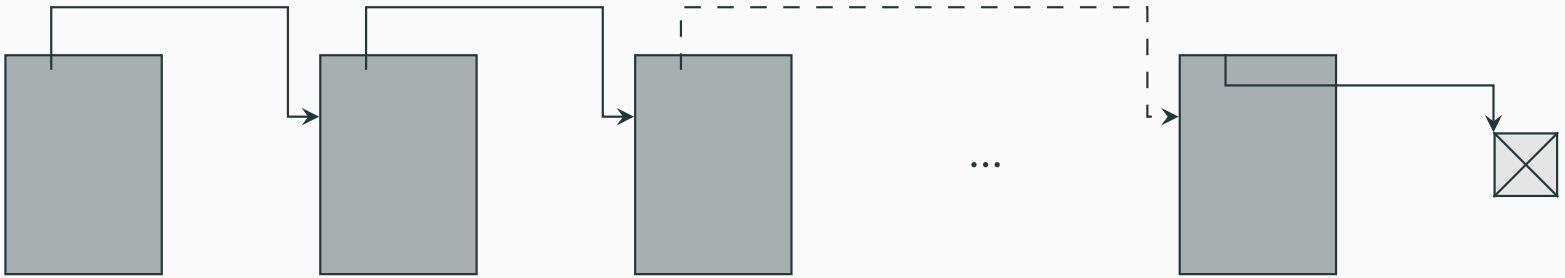
xv6 memory management [1]



- A linked-list of free pages
- Constant allocation time - $O(1)$

[1] kalloc.c

xv6 memory management [1]



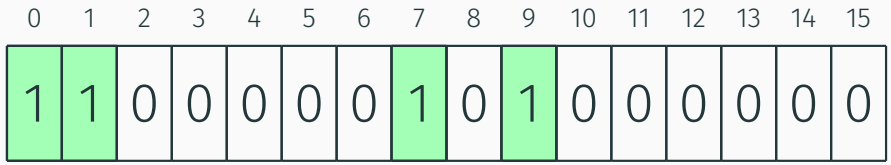
- A linked-list of free pages
- Constant allocation time - $O(1)$

Problem?

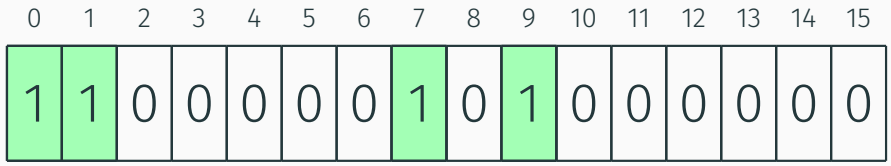
- **Memory wastage:** We use 4KiB for storing a pointer (4 bytes)

[1] kalloc.c

Bitmap allocator

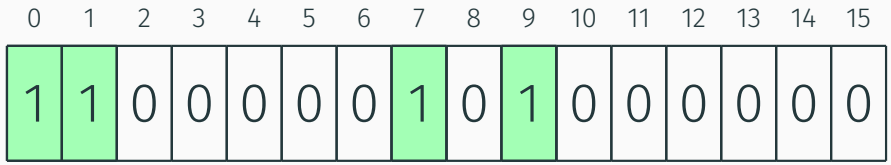


Bitmap allocator



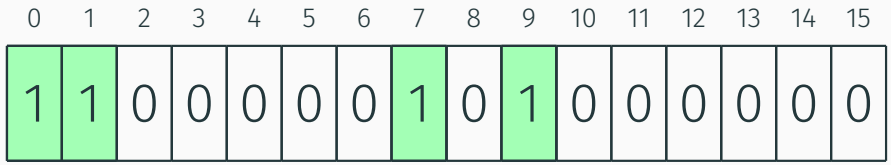
- Use a single-bit for a page (0 - occupied, 1 - free)

Bitmap allocator



- Use a single-bit for a page (0 - occupied, 1 - free)
- Linear allocation time - $O(n)$

Bitmap allocator



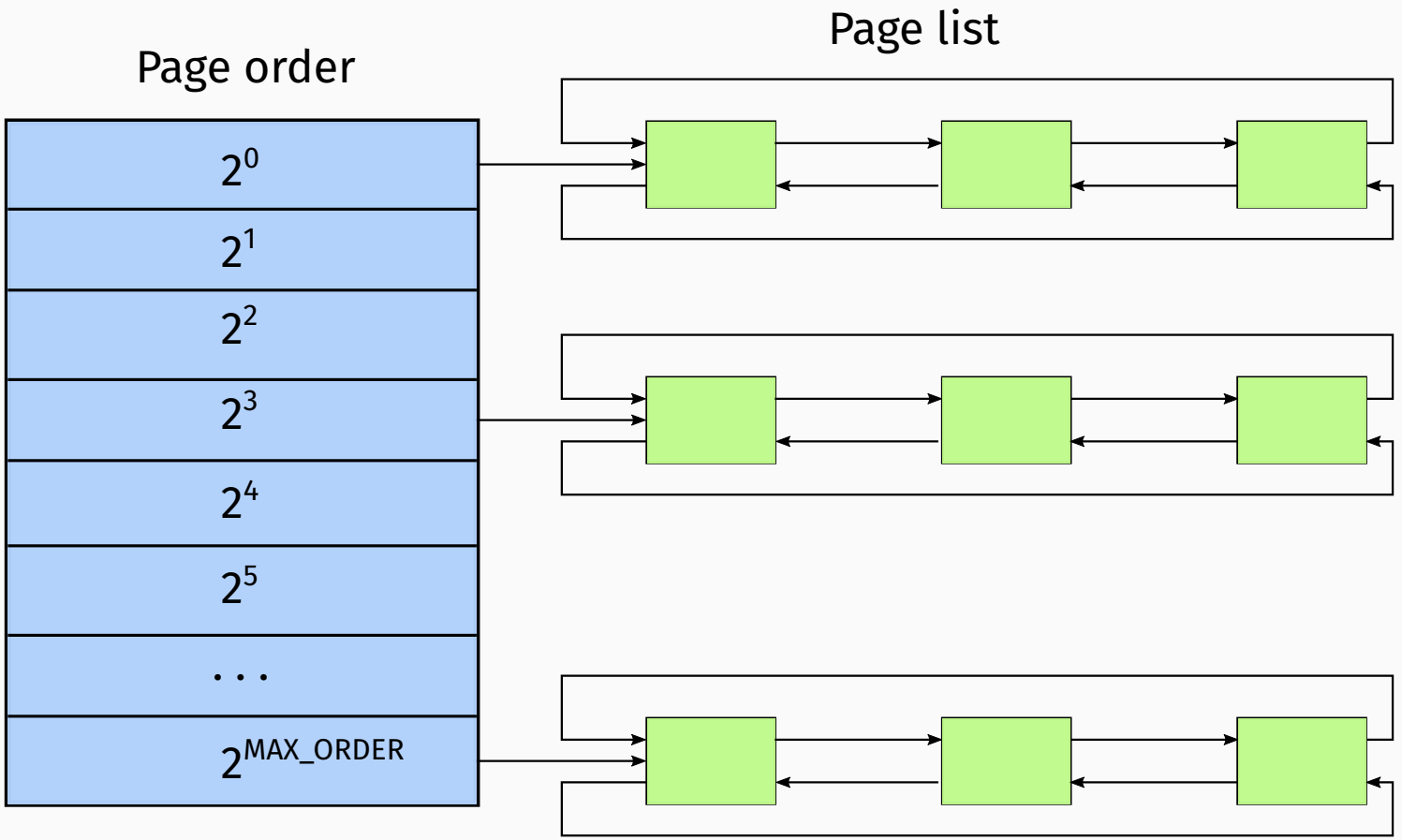
- Use a single-bit for a page (0 - occupied, 1 - free)
- Linear allocation time - $O(n)$

Problems?

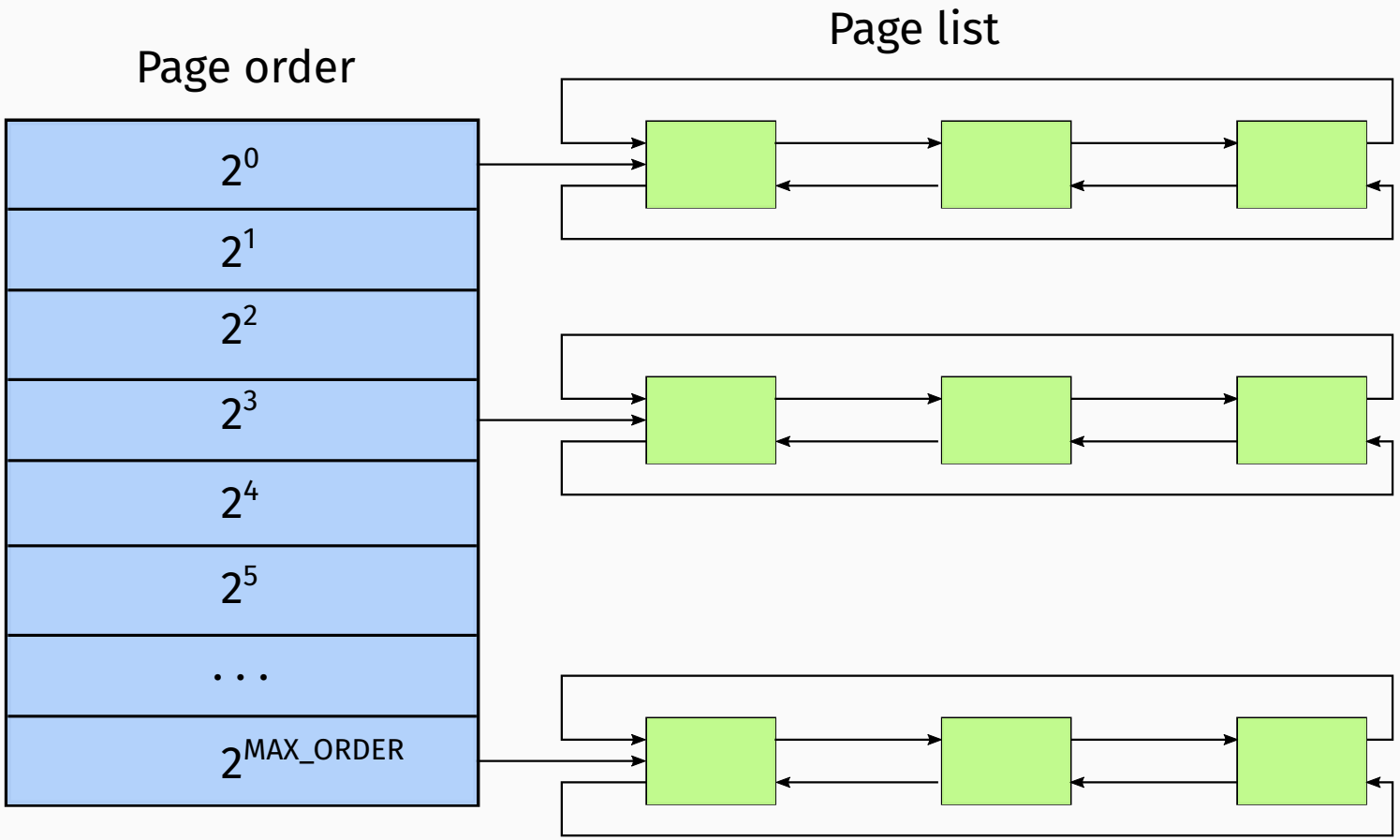
- **External Fragmentation:** Inability to service a request despite having free memory
- Linear scan takes time

Buddy allocator

Buddy allocator



Buddy allocator



- Maintains a list of blocks of various sizes

Buddy allocator - alloc/free

- Allocation strategy - split

Buddy allocator - alloc/free

- Allocation strategy - split
  - Find a free block in the desired list

Buddy allocator - alloc/free

- Allocation strategy - split
  - Find a free block in the desired list
  - If not, split a higher order block size ($2^2 = 2^1 \cdot 2^1$)

Buddy allocator - alloc/free

- Allocation strategy - split
  - Find a free block in the desired list
  - If not, split a higher order block size ($2^2 = 2^1 \cdot 2^1$)
- Free strategy - coalesce

Buddy allocator - alloc/free

- Allocation strategy - split
  - Find a free block in the desired list
  - If not, split a higher order block size ($2^2 = 2^1 \cdot 2^1$)
- Free strategy - coalesce
  - Coalesce if the buddy block is free

Buddy allocator - alloc/free

- Allocation strategy - split
  - Find a free block in the desired list
  - If not, split a higher order block size ($2^2 = 2^1 \cdot 2^1$)
- Free strategy - coalesce
  - Coalesce if the buddy block is free

Problem?

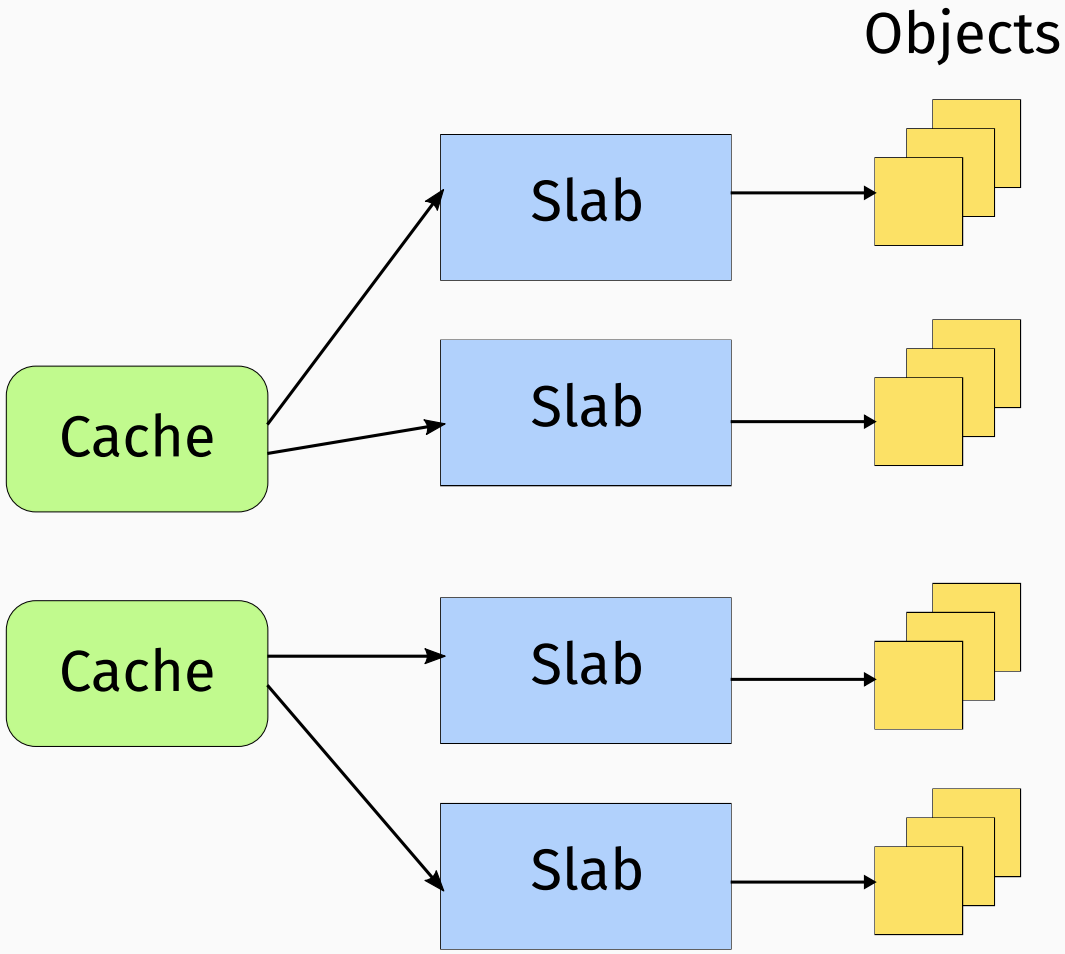
Internal Fragmentation: Minimum allocation unit is a page

Slab allocator

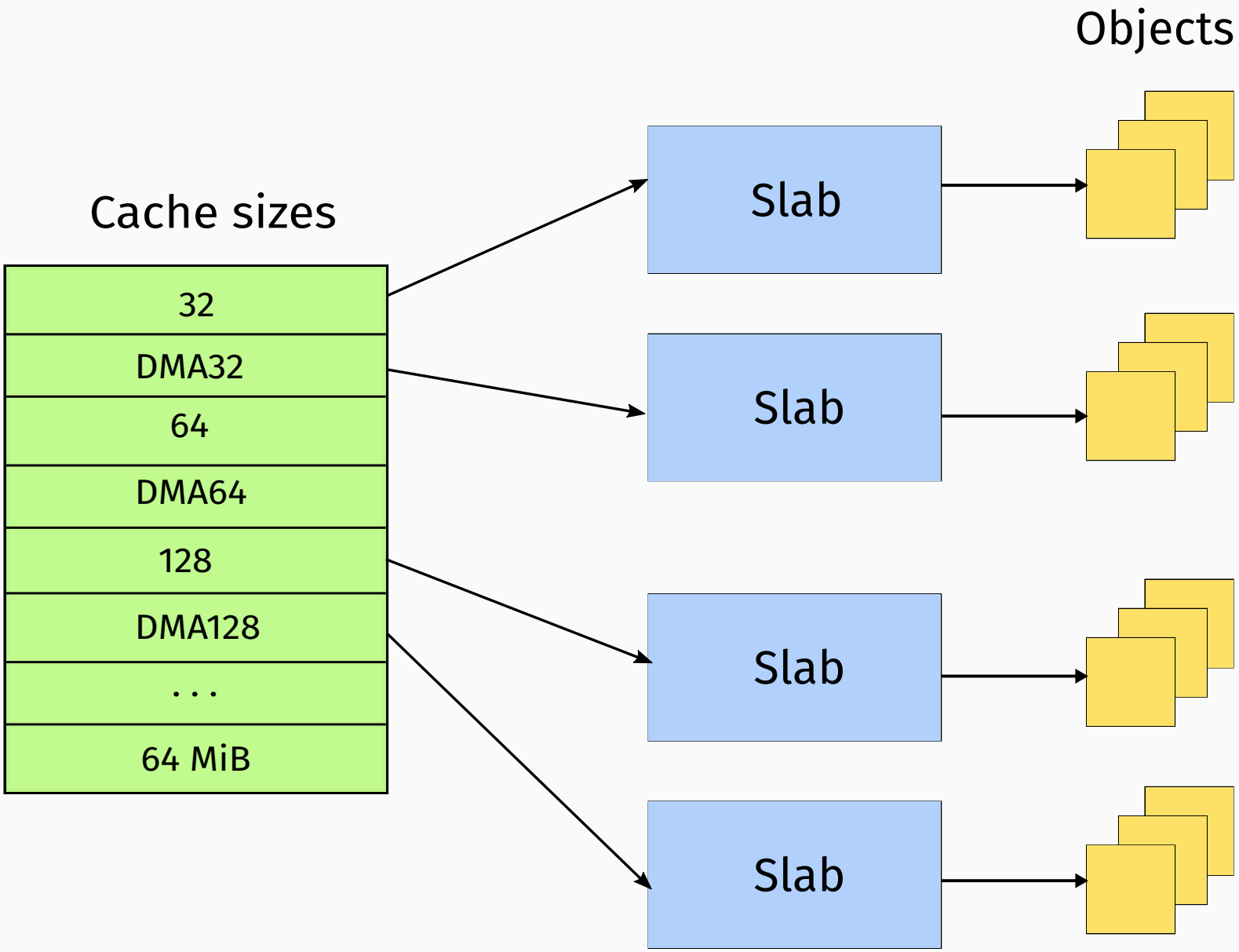
Slab allocator

- Solve internal fragmentation - allocate objects of any size
- Maintain an object cache - to save time
- Align object to hardware cacheline boundaries - for efficiency

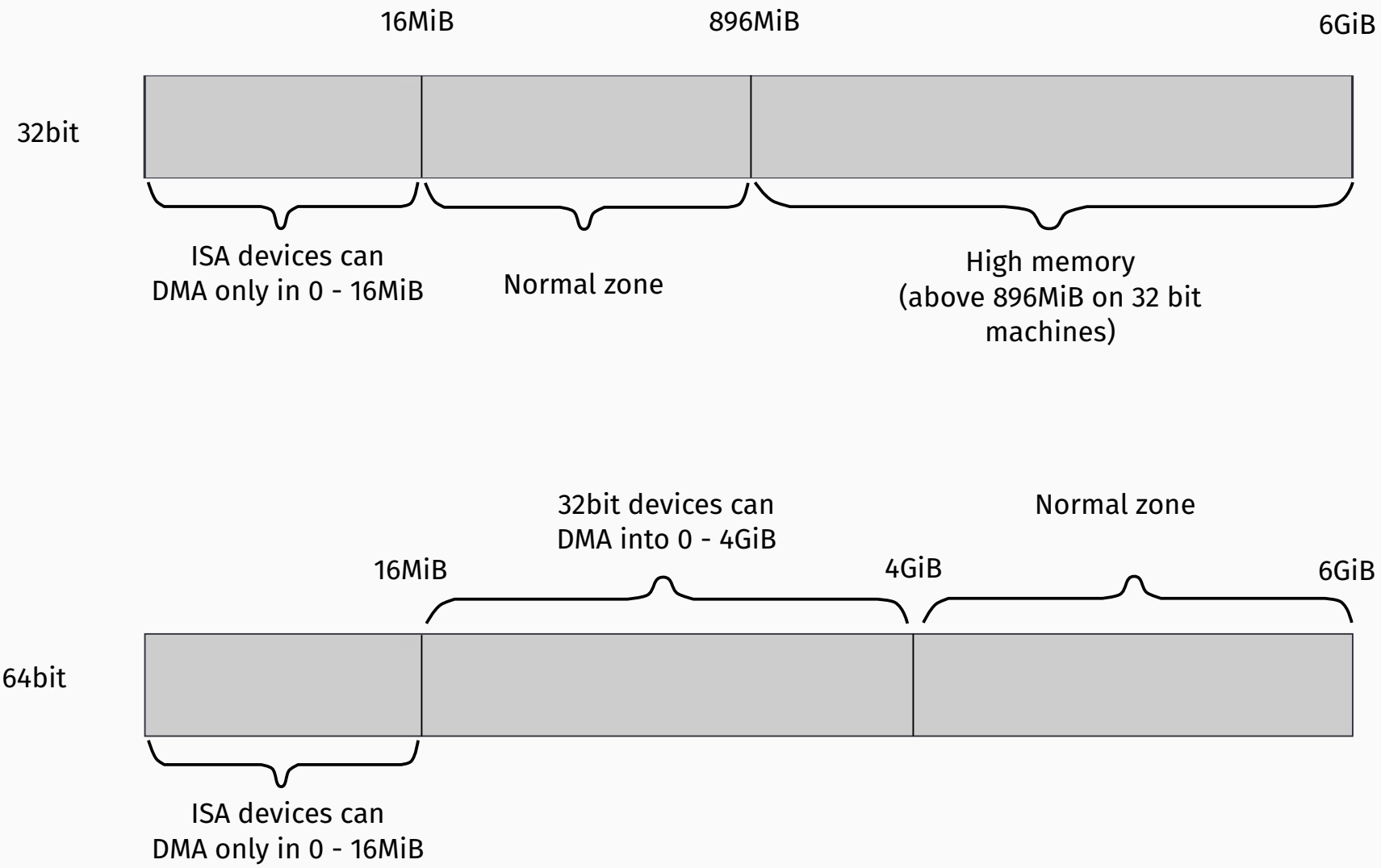
Slab allocator



kmalloc - table of caches

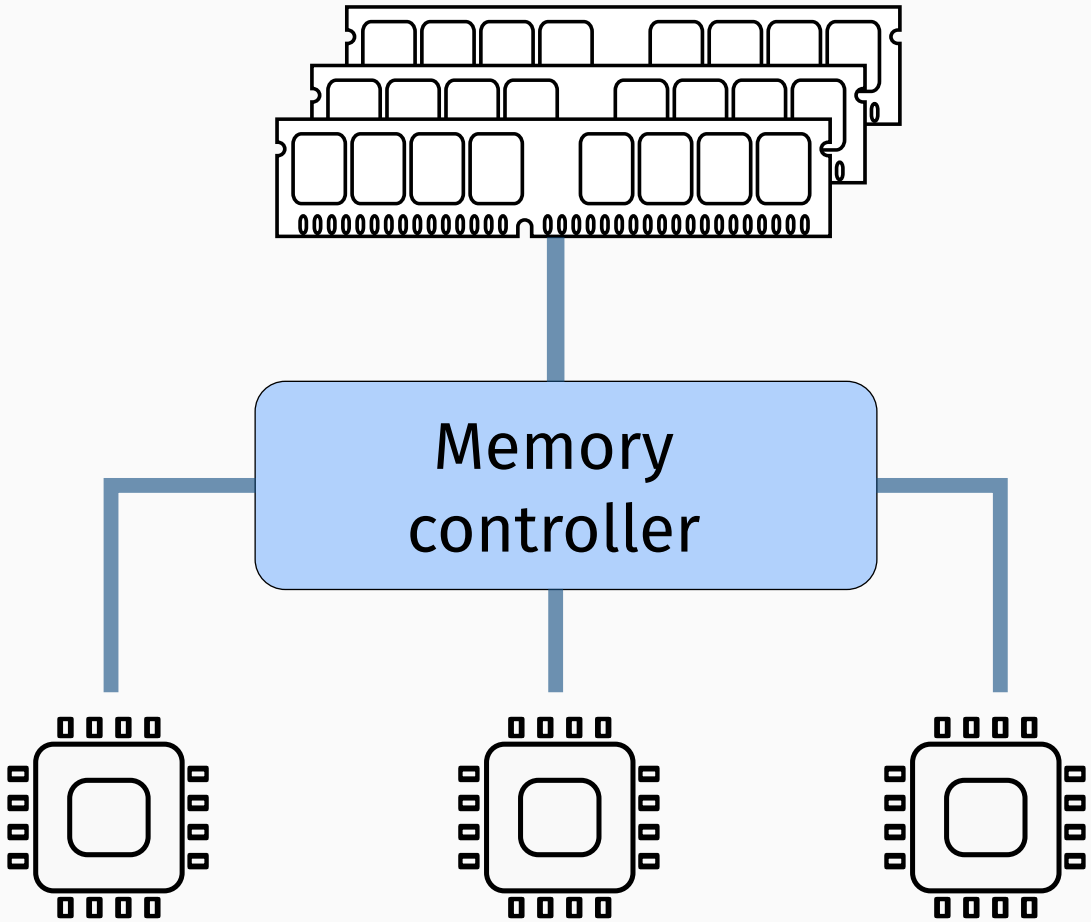


Zones

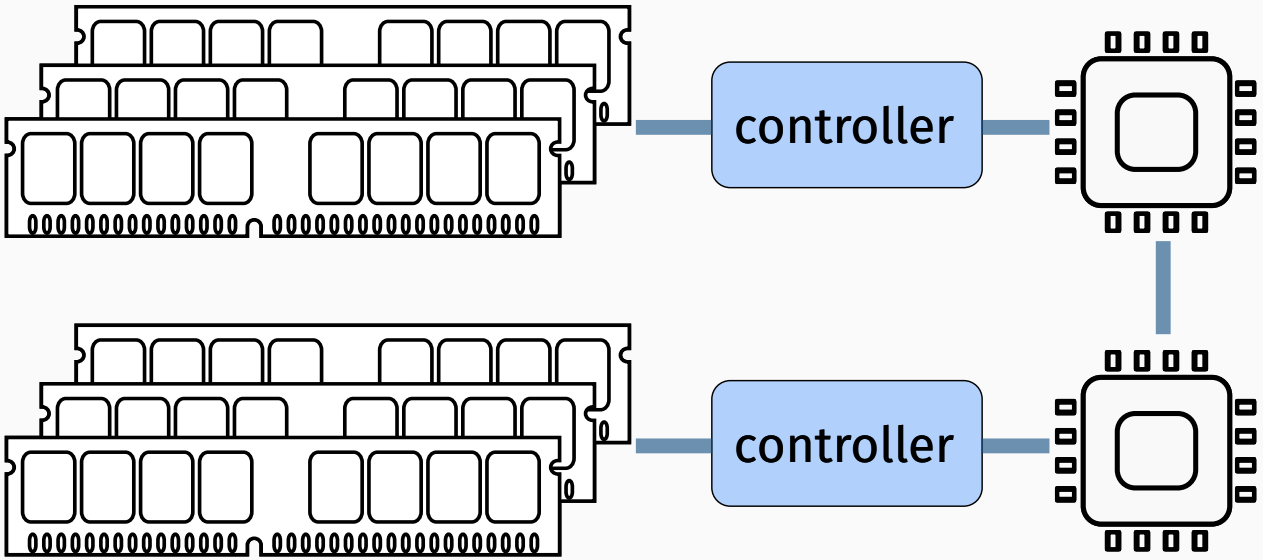

16MiB
896MiB
6GiB
32bit
ISA devices can
DMA only in 0 - 16MiB
Normal zone
High memory
(above 896MiB on 32 bit
machines)
32bit devices can
DMA into 0 - 4GiB
Normal zone
16MiB
4GiB
6GiB
64bit
ISA devices can
DMA only in 0 - 16MiB

Memory architectures

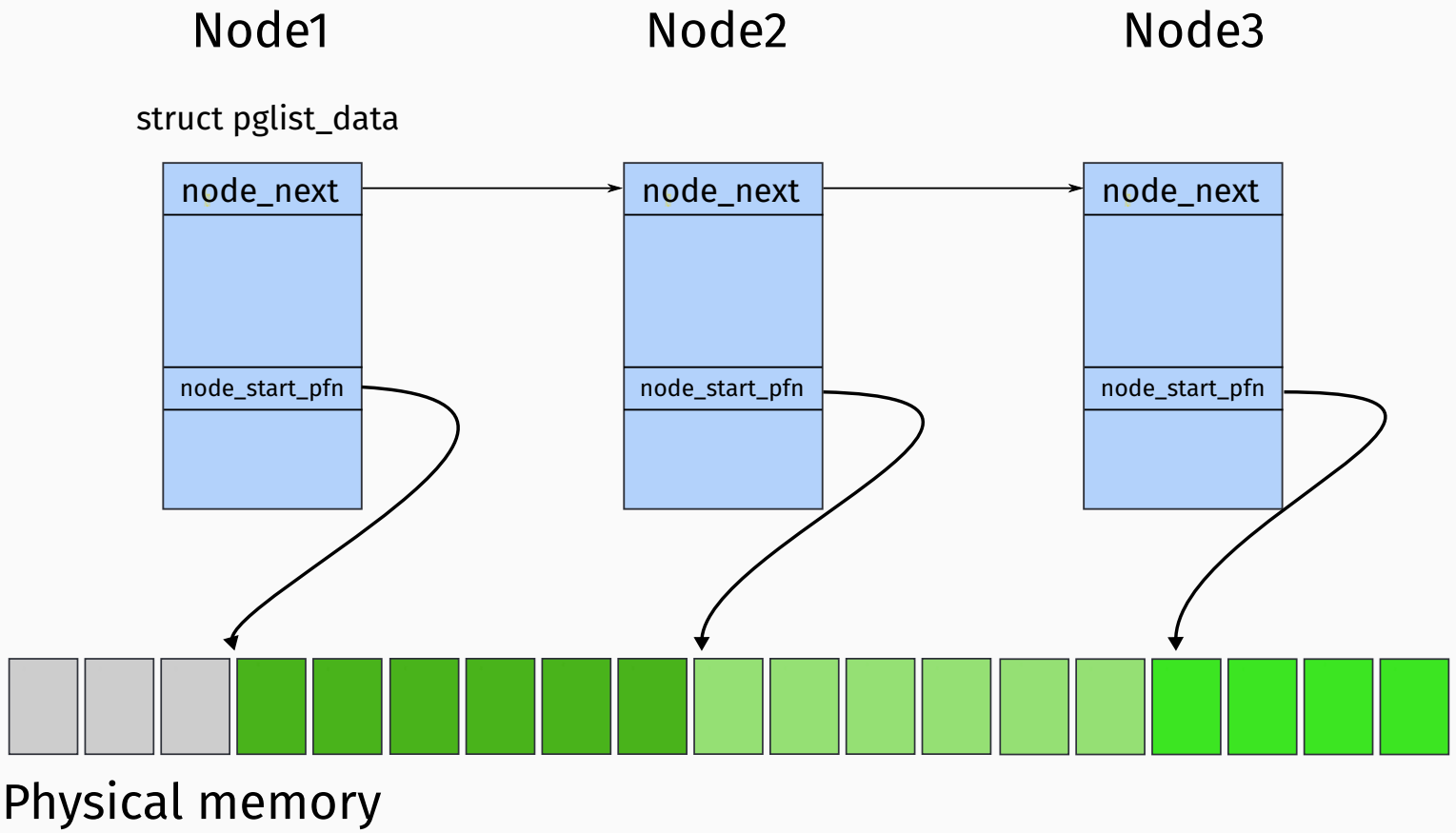
Uniform memory access(UMA)



Non-Uniform memory access(NUMA)



Nodes


Node1
Node2
Node3
struct pglist_data
node_next
node_start_pfn
node_next
node_start_pfn
node_next
node_start_pfn
Physical memory

Memory organization hierarchy (Linux)

- Nodes (`struct pglist_data`)
  - Abstraction of memory on each memory controller
- Zonelists (`struct zonelists`)
  - List of fallback zones in other nodes - if allocation fails
- Zones (`struct zone`)
  - x86 - `ZONE_DMA, ZONE_NORMAL, ZONE_HIGHMEM`
  - x64 - `ZONE_DMA, ZONE_DMA32, ZONE_NORMAL`
- Pages (`struct page`)
  - Managed by buddy allocator
- Small-sized objects
  - Slab allocator (`struct kmem_cache`), `kmalloc`

Thank you!