CS238P: Operating Systems

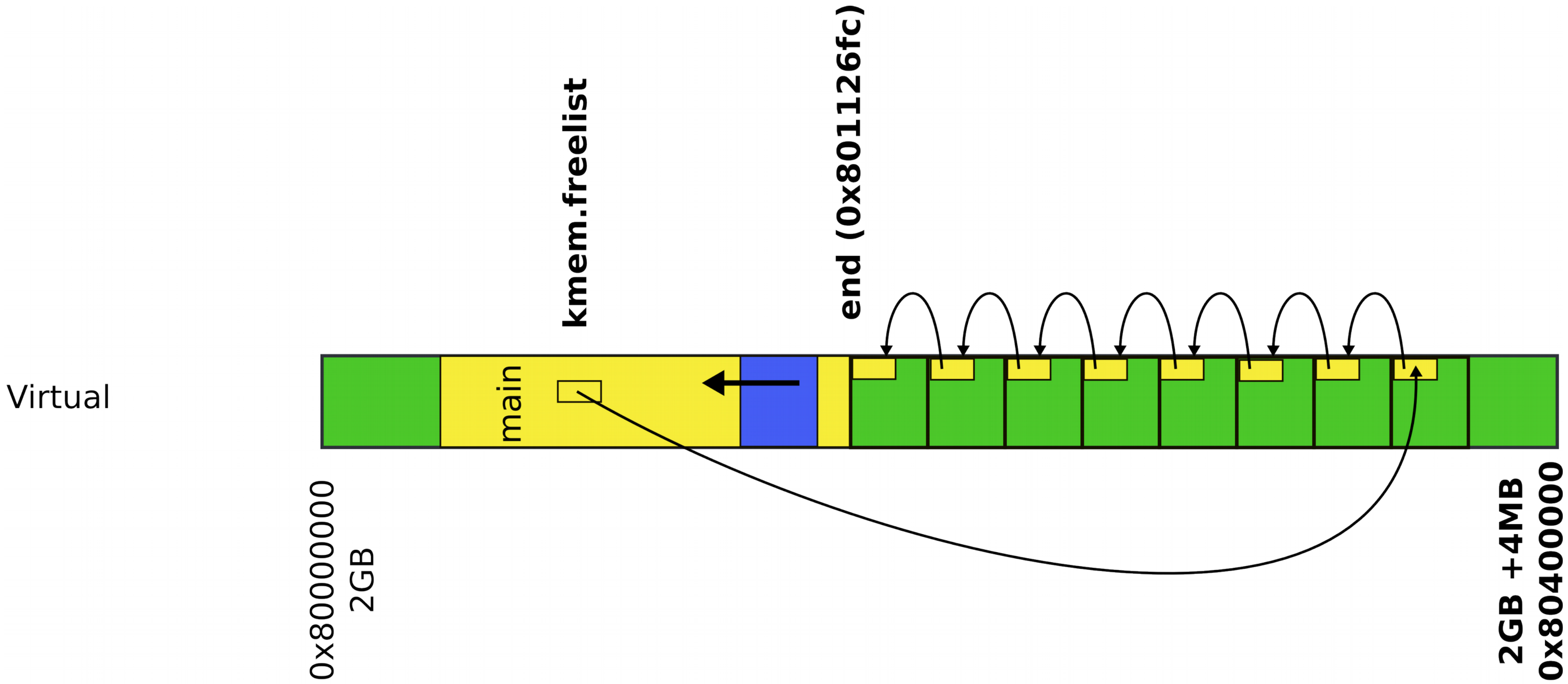
Lecture 13: Memory management

Anton Burtsev
November, 2018

Xv6 Book, Chapter 1. KERNBASE limits the amount of memory a single process can use, which might be irritating on a machine with a full 4 GB of RAM.

Would raising KERNBASE allow a process to use more memory?

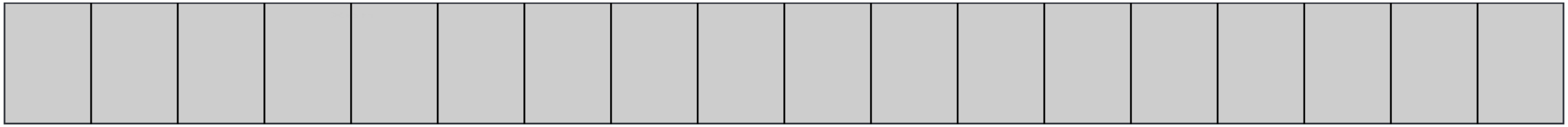
Xv6: physical page allocator



```
CS : 0x8             EIP: main
SS : 0x10            ESP: stack
GDT: 0x7c78          TSS: 0x0
IDT: 0x0             CR3: entrypgdir
```

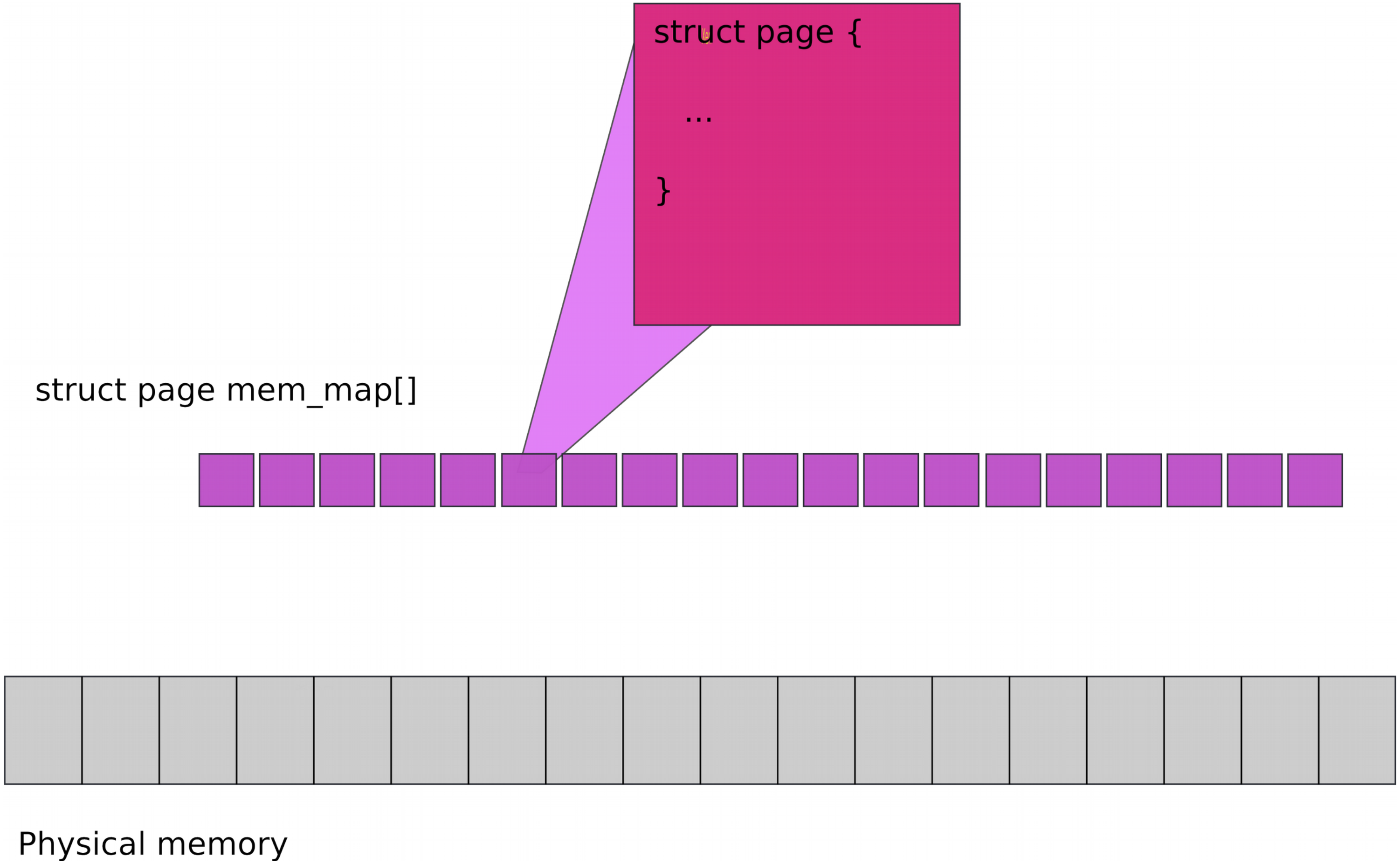
Protected Mode

Physical memory



Physical memory

We need a smaller array to describe physical pages, e.g., mem_map[] in Linux



Memory allocation

Simplest memory allocator

- Bitmap of all pages
  - Bootmem allocator in Linux
- Allocation searches for an unused page
  - Multiple sub-page allocations can be served from the same page by advancing a pointer

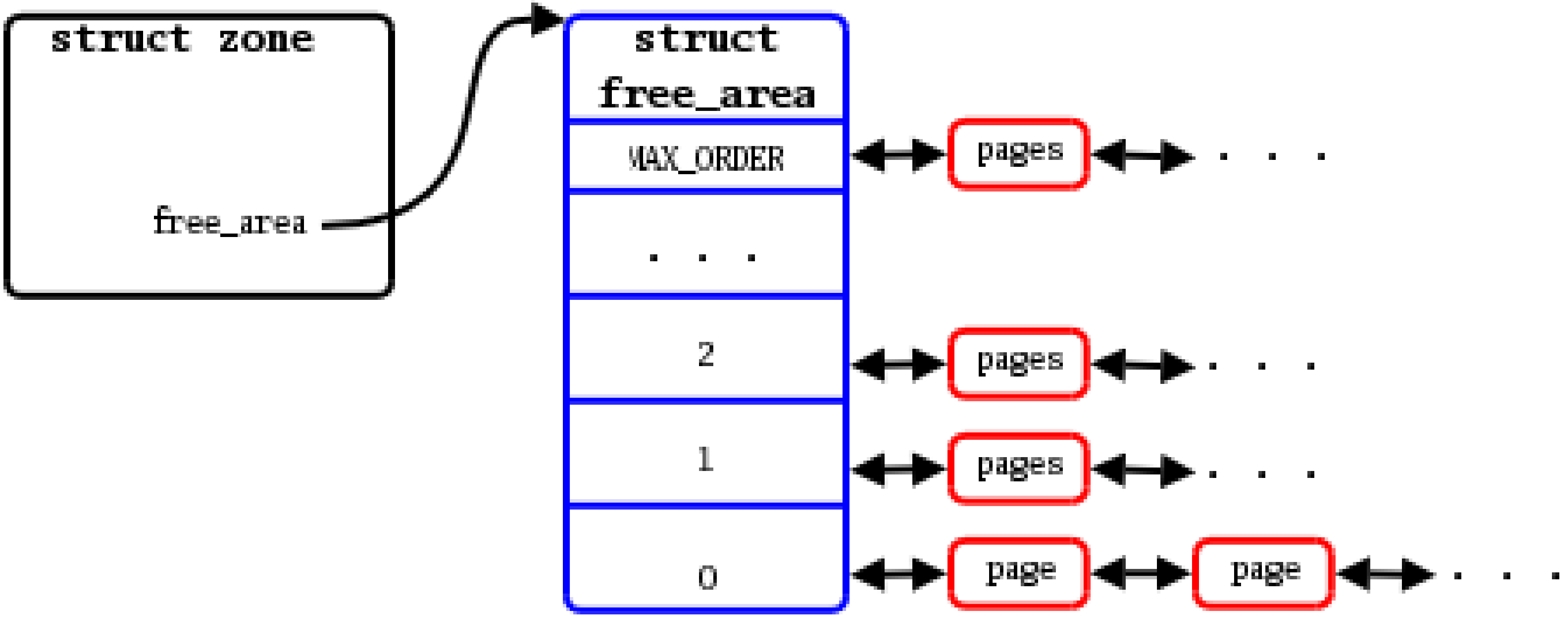
- Works ok, but what is the problem?

Boot memory allocator

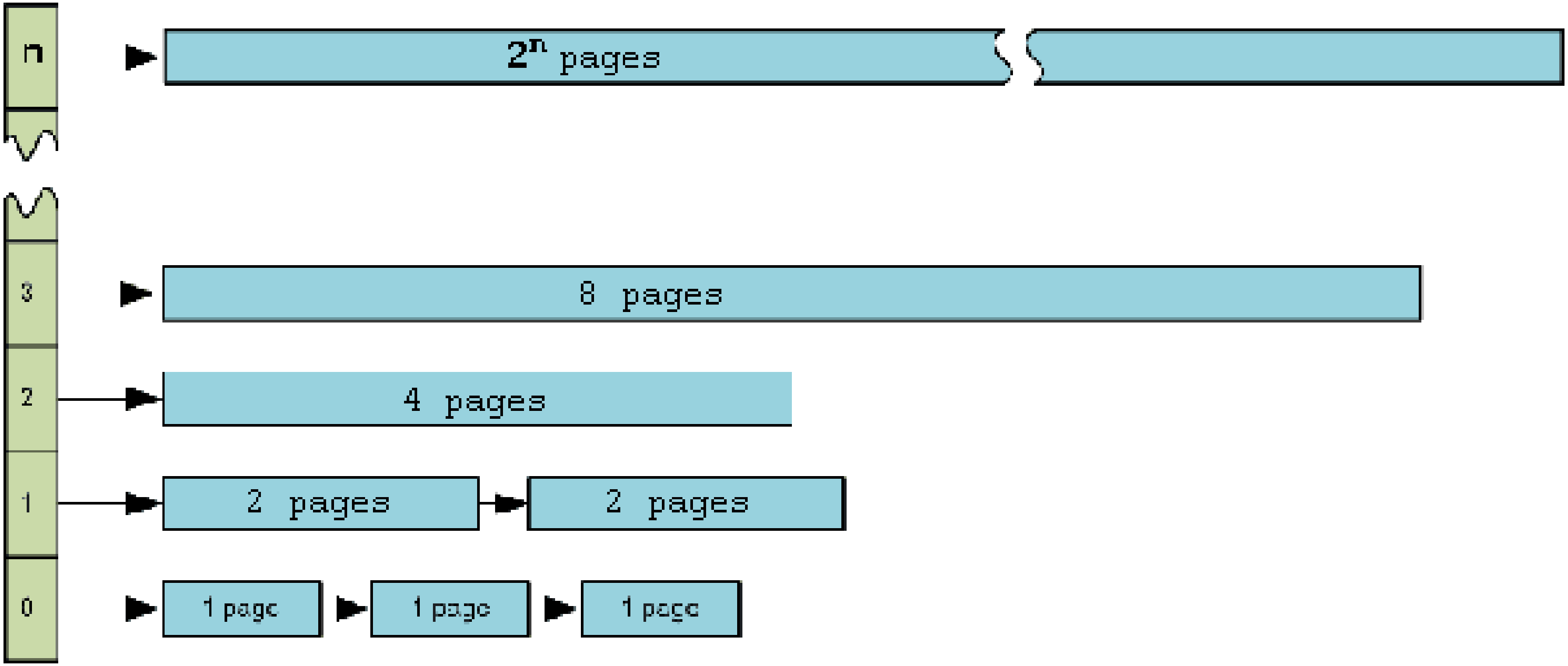
- Bitmap of all pages
  - Bootmem allocator in Linux
- Allocation searches for an unused page
  - Multiple sub-page allocations can be served from the same page by advancing a pointer
- Works ok, but what is the problem?
  - Linear scan of the bitmap
    - Too long

Buddy:
Physical Memory Allocator

Buddy memory allocator



Buddy allocator



What's wrong with buddy?

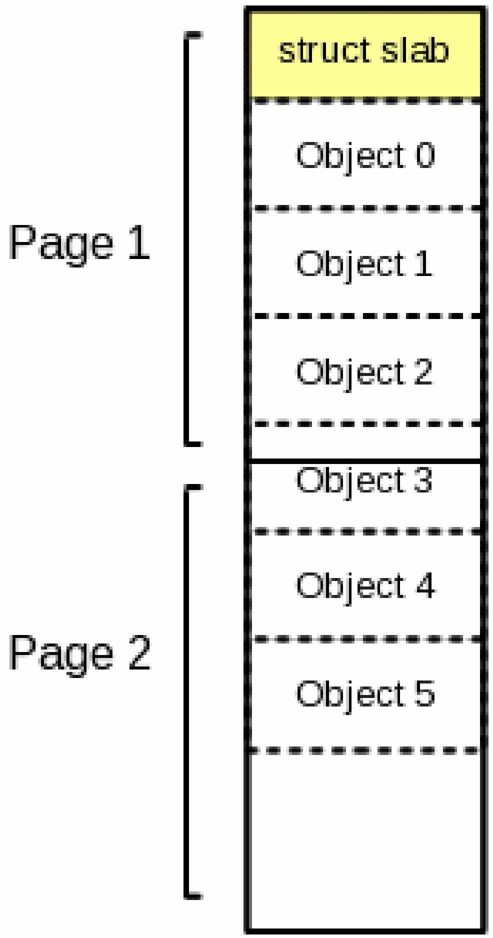
What's wrong with buddy?

- Buddy allocator is ok for large allocations
  - E.g. 1 page or more
- But what about small allocations?
  - Buddy uses the whole page for a 4 bytes allocation
    - Wasteful
  - Buddy is still slow for short-lived objects

Slab:
Allocator for object of a fixed size

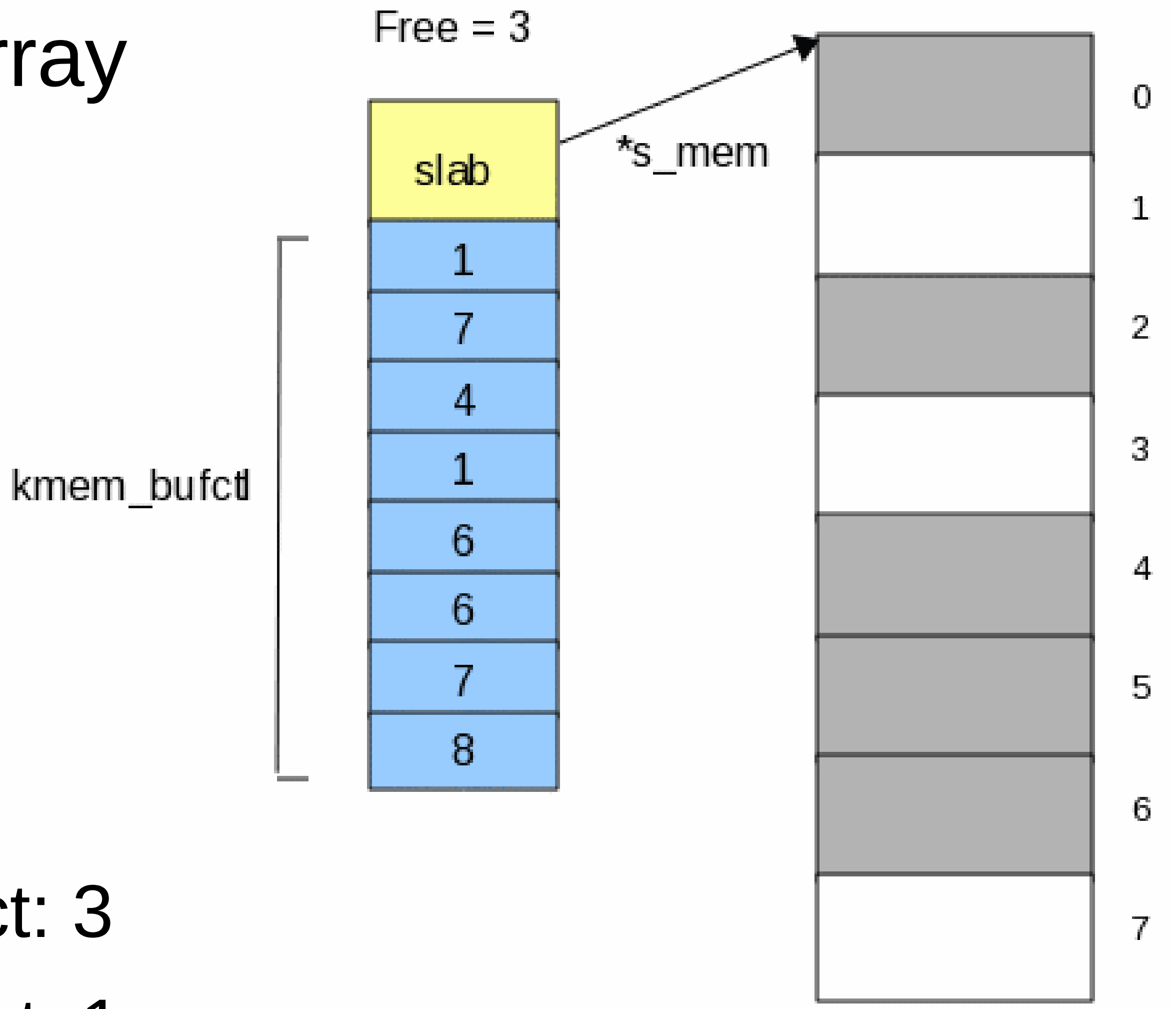
Slab

- A 2 page slab with 6 objects



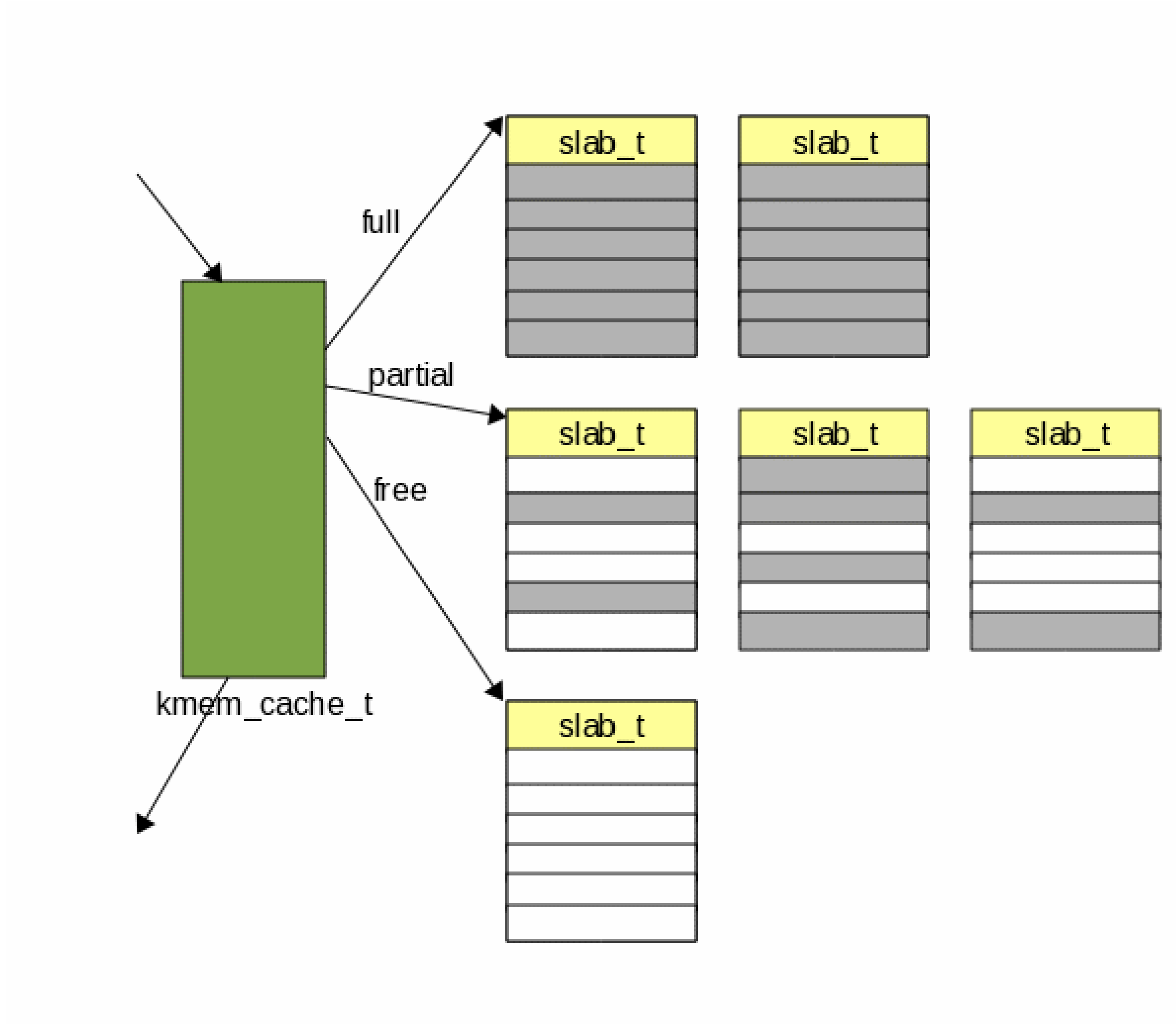
Keeping track of free objects

- kmem_bufctl array is effectively a linked list



- First free object: 3
- Next free object: 1

A cache is formed out of slabs



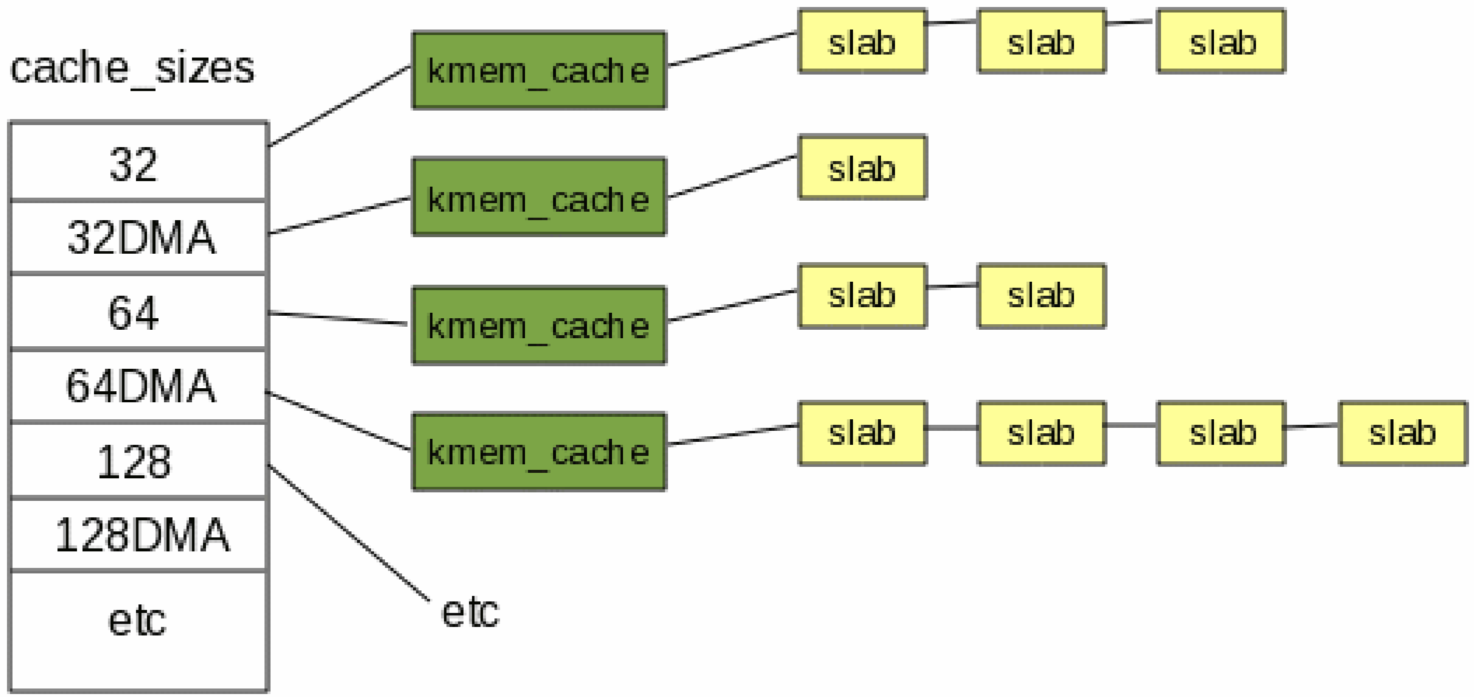
Slab is fine, but what's wrong?

Slab is fine, but what's wrong?

- We can only allocate objects of one size

Kmalloc(): variable size objects

- A table of caches
  - Size: 32, 64, 128, etc.

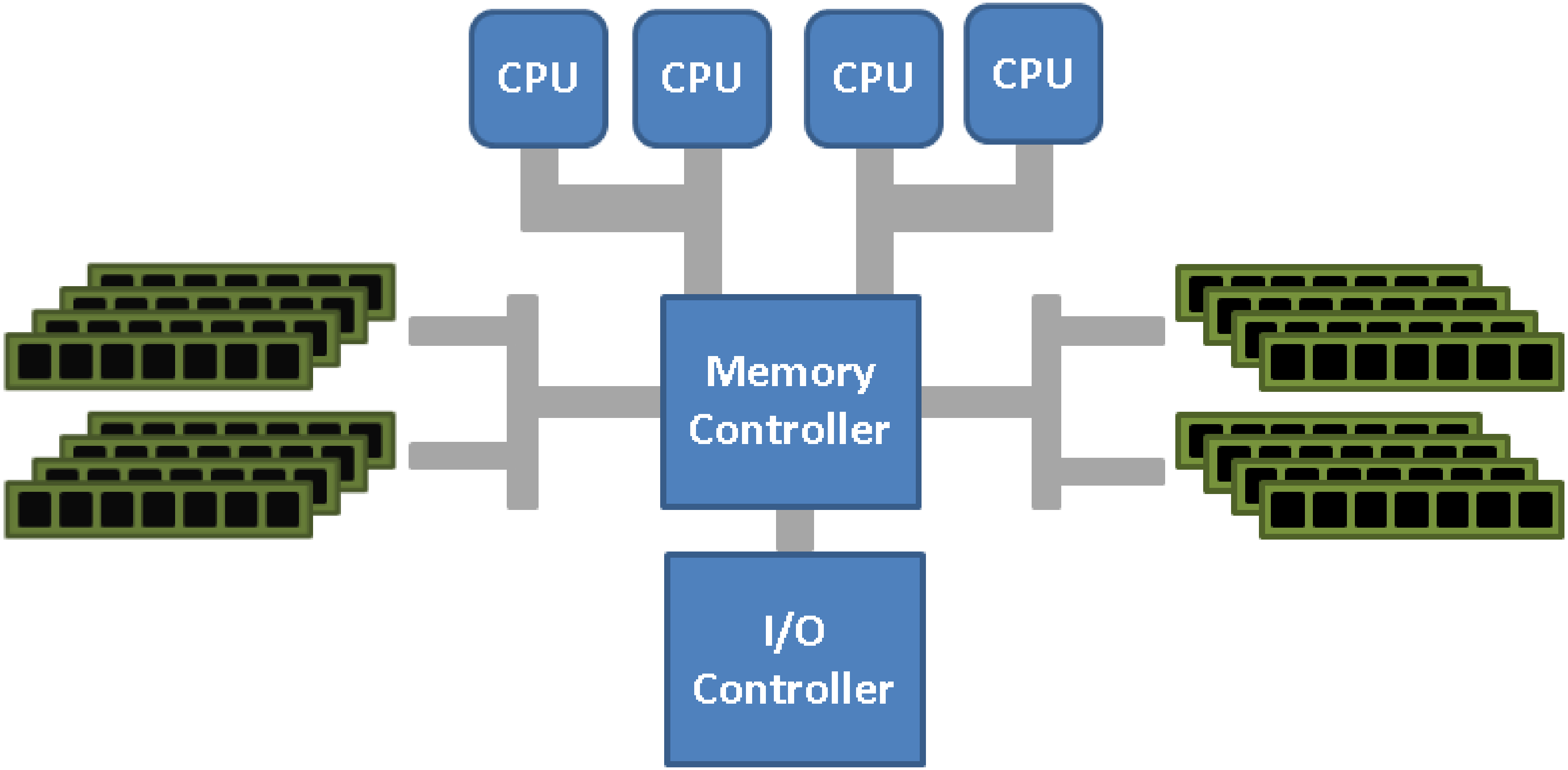


NUMA
Non-uniform memory access

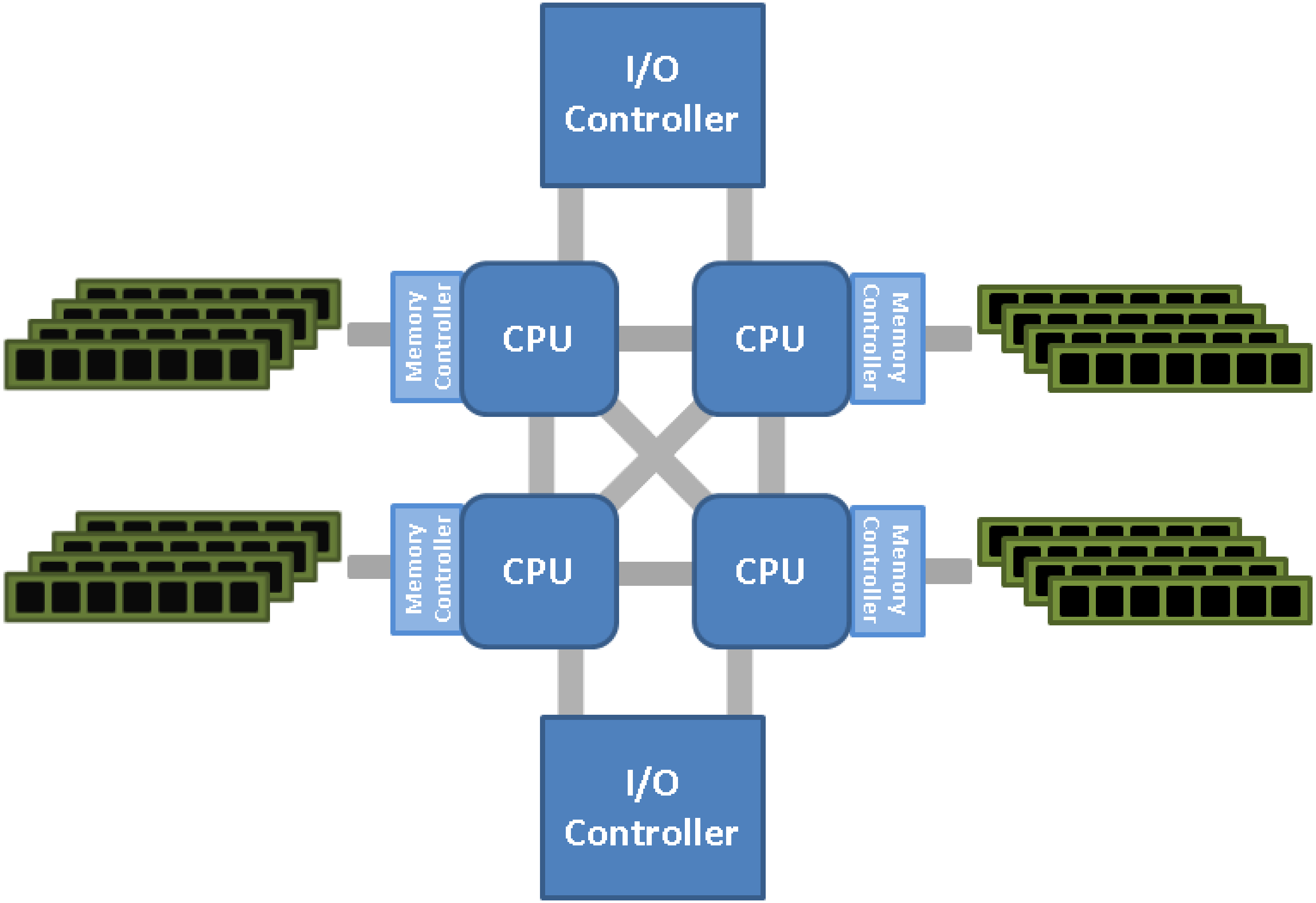
Uniform and non-uniform memory access

- Parts of memory can be faster than others

Uniform memory access (UMA)

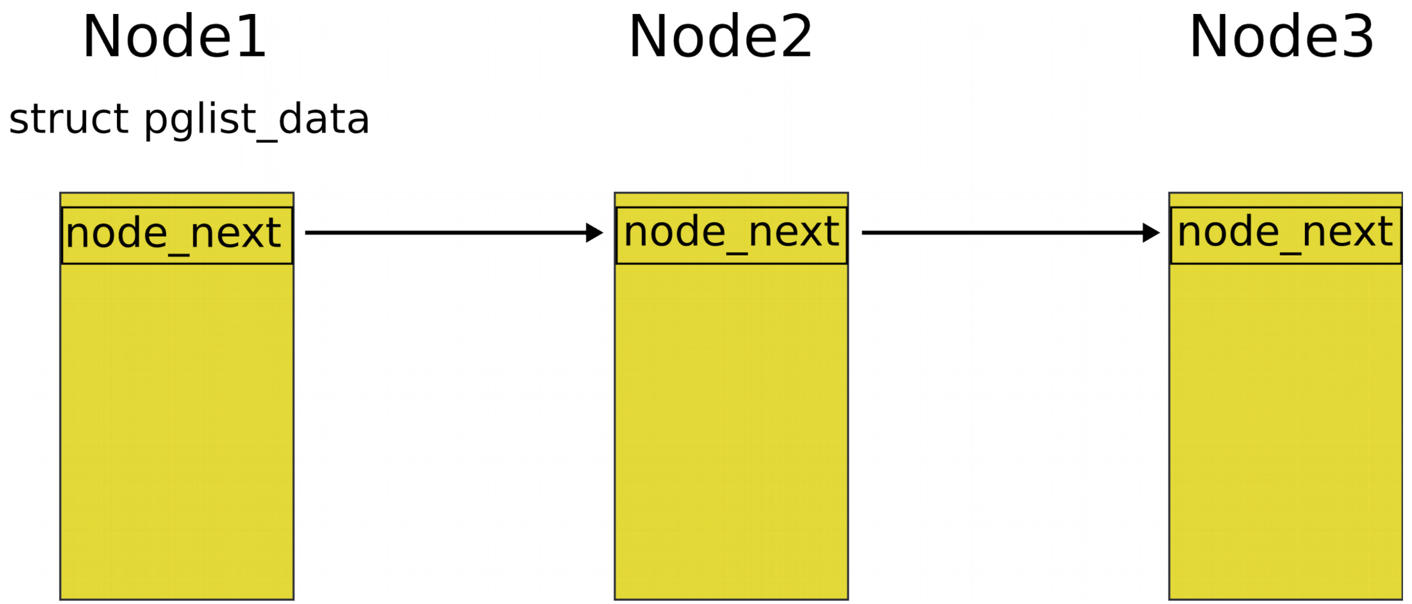


Nonuniform memory access (NUMA)

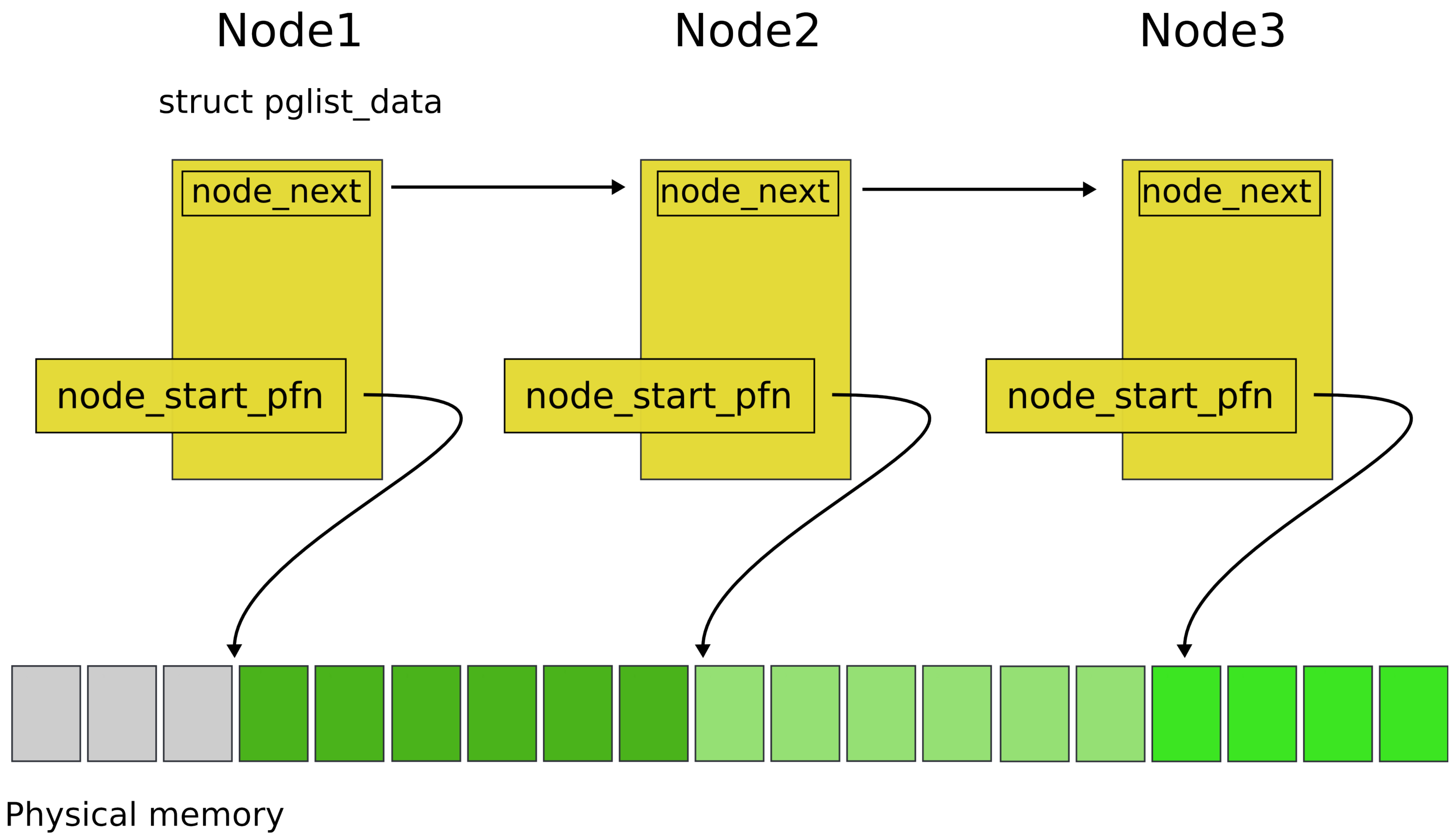


Nodes

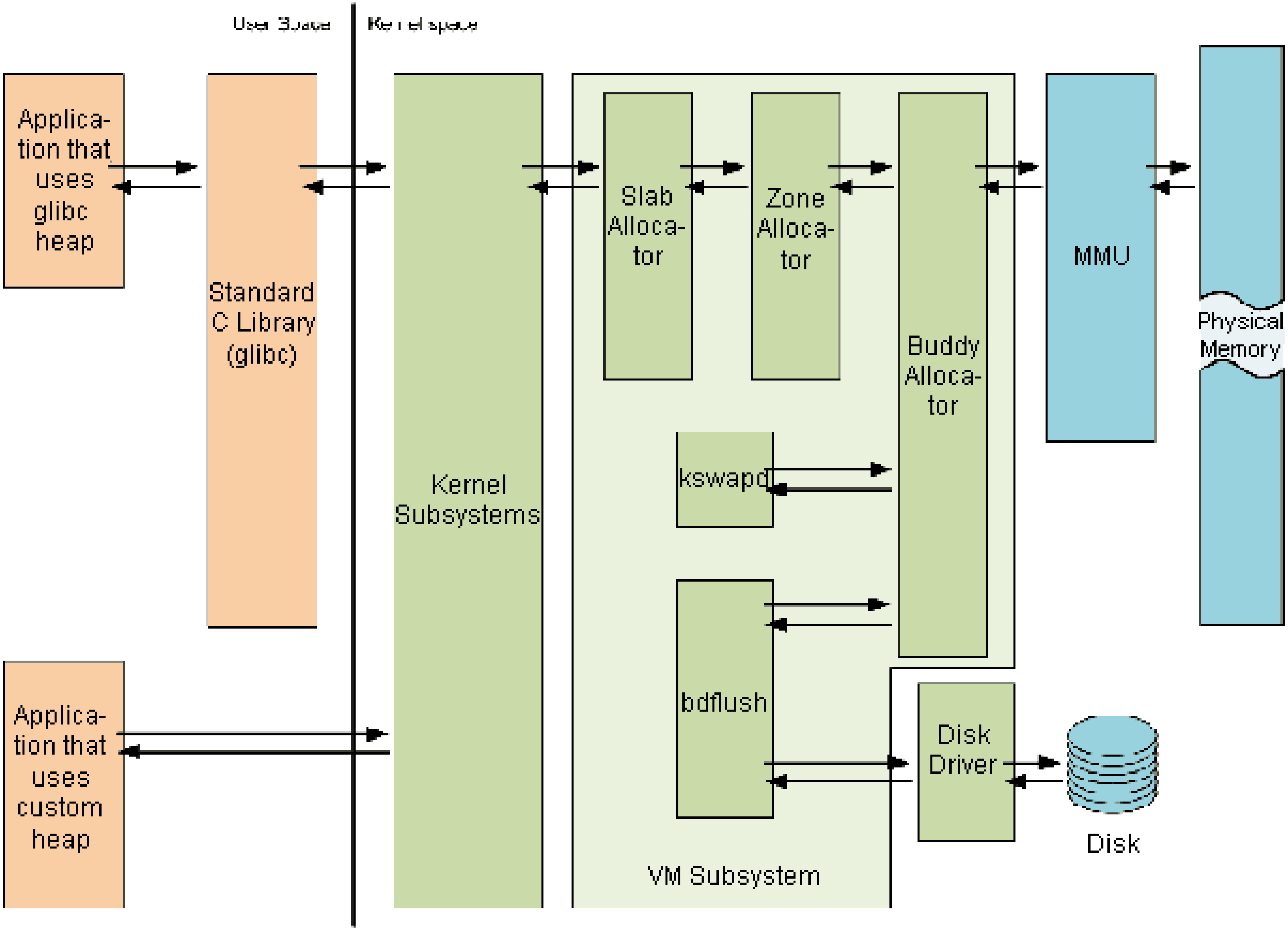
- Attempt to allocate memory from the current node
  - Fall back to the next node in list
    - If ran out of local memory



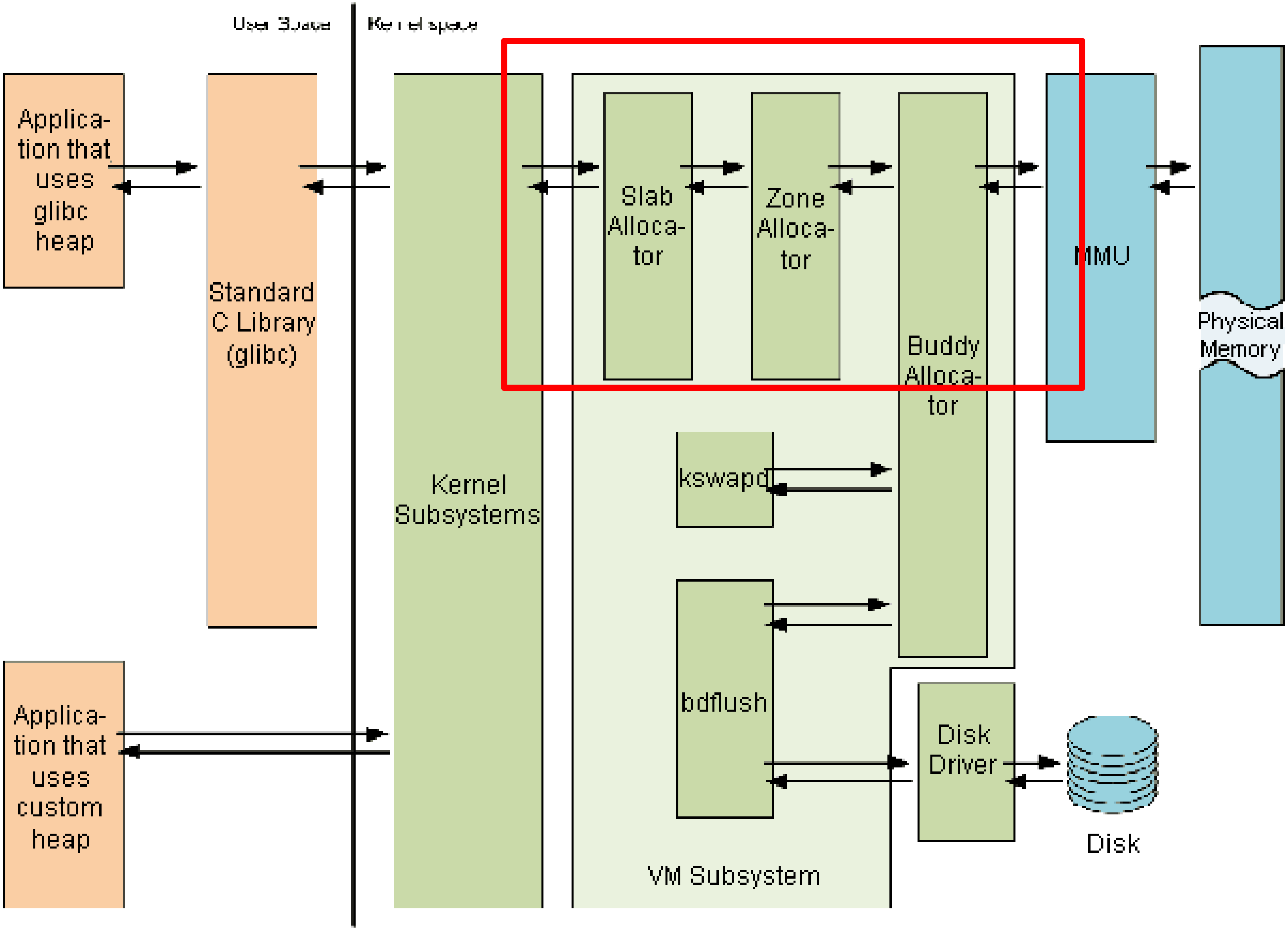
Nodes


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

Linux memory management



Linux memory management



User Space

Kernel space

Application that uses glibc heap

Standard C Library (glibc)

Kernel Subsystems

Slab Allocator

Zone Allocator

Buddy Allocator

MMU

Physical Memory

kswapd

bdflush

Disk Driver

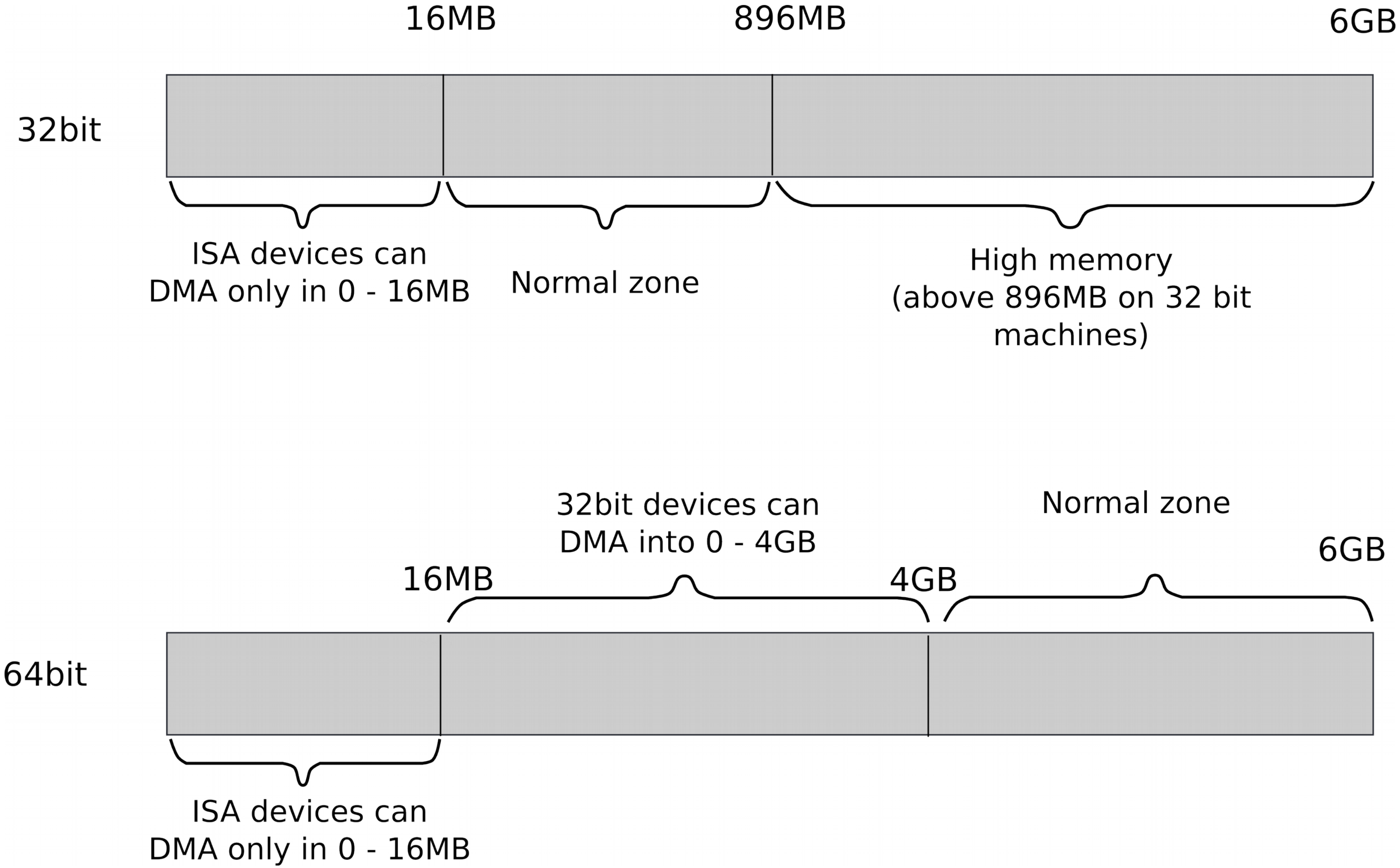
Disk

VM Subsystem

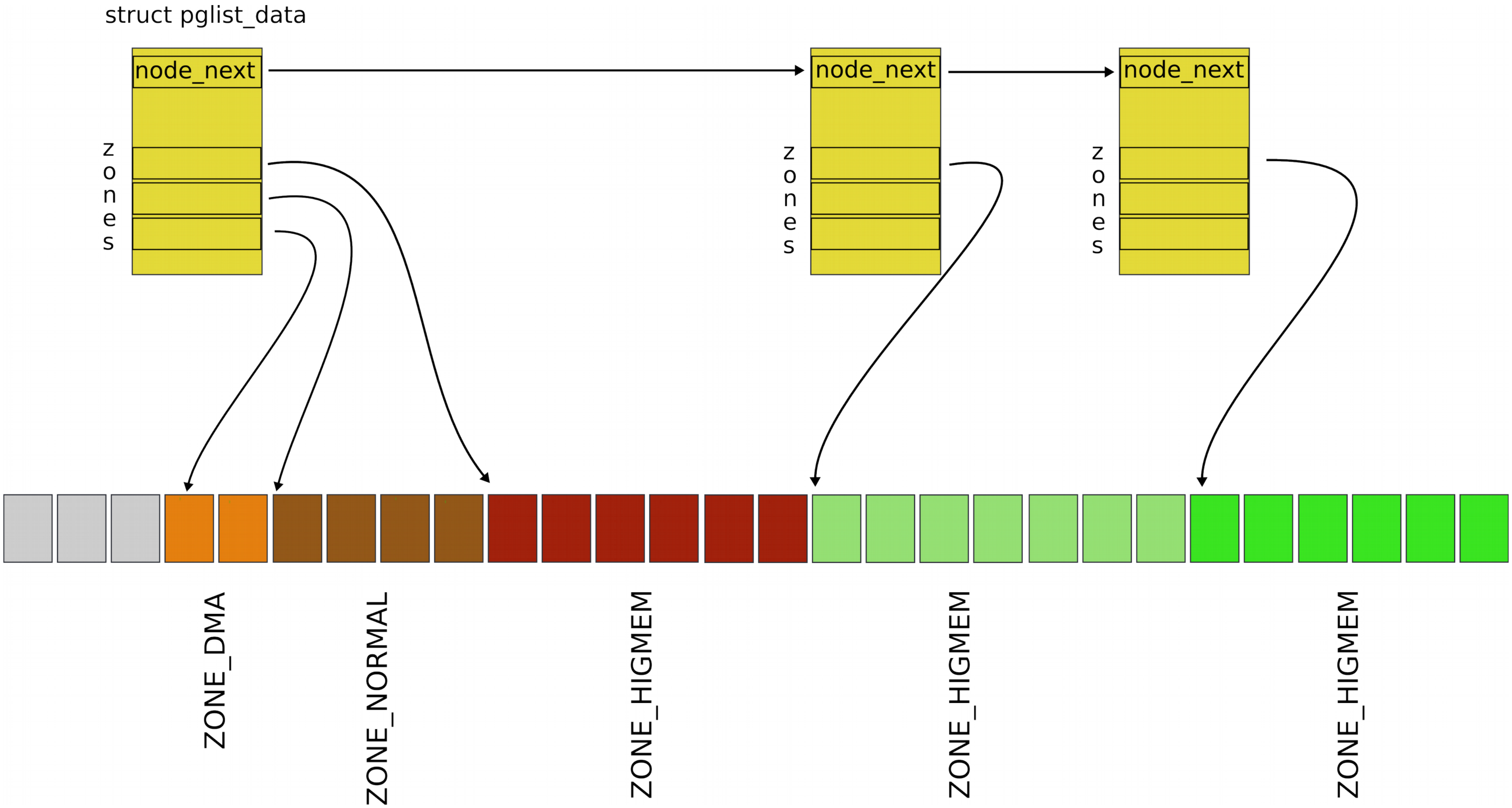
Application that uses custom heap

Thank you!

Zones


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

Zones


struct pglist_data
node_next
node_next
node_next
zones
zones
zones
ZONE_DMA
ZONE_NORMAL
ZONE_HIGMEM
ZONE_HIGMEM
ZONE_HIGMEM