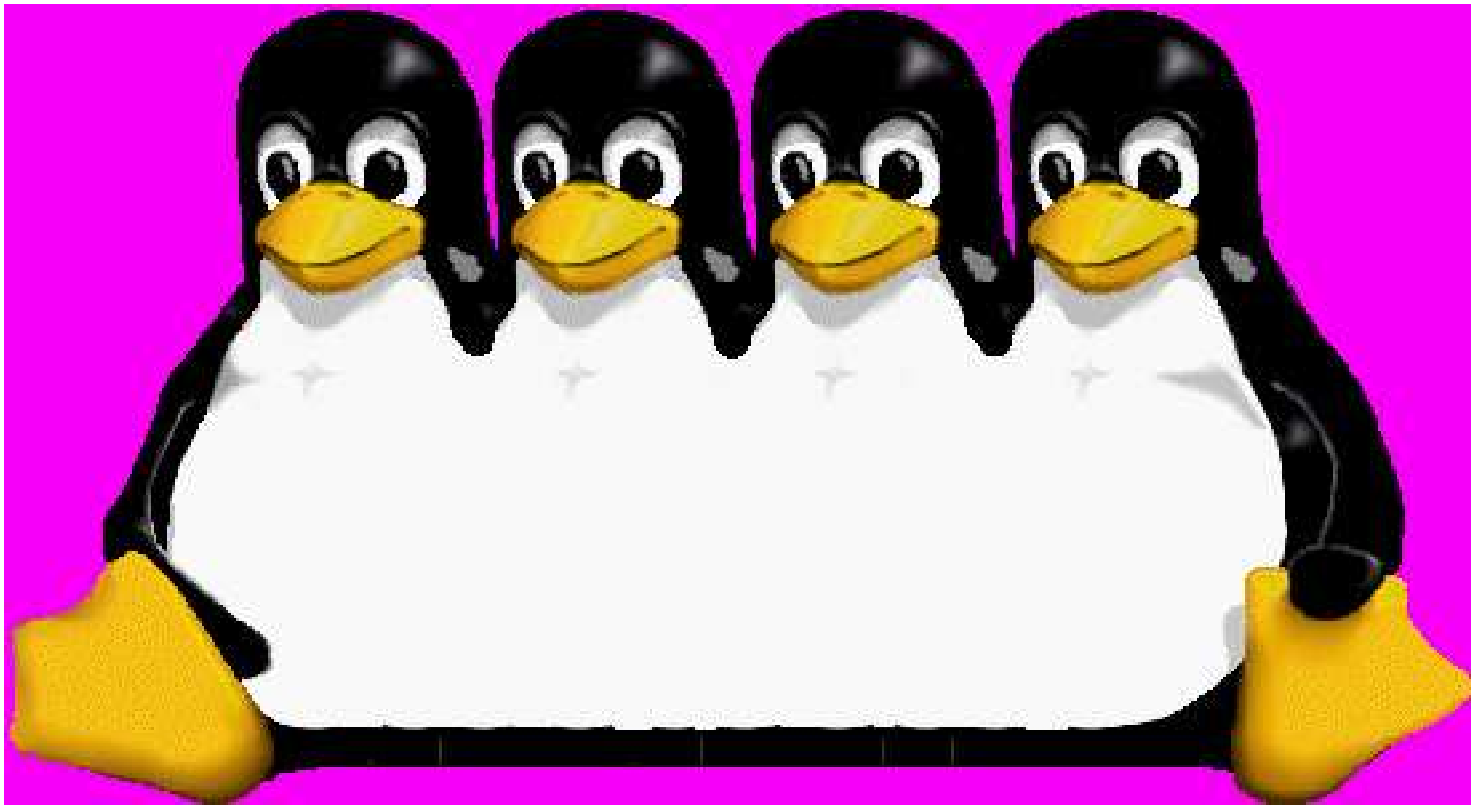
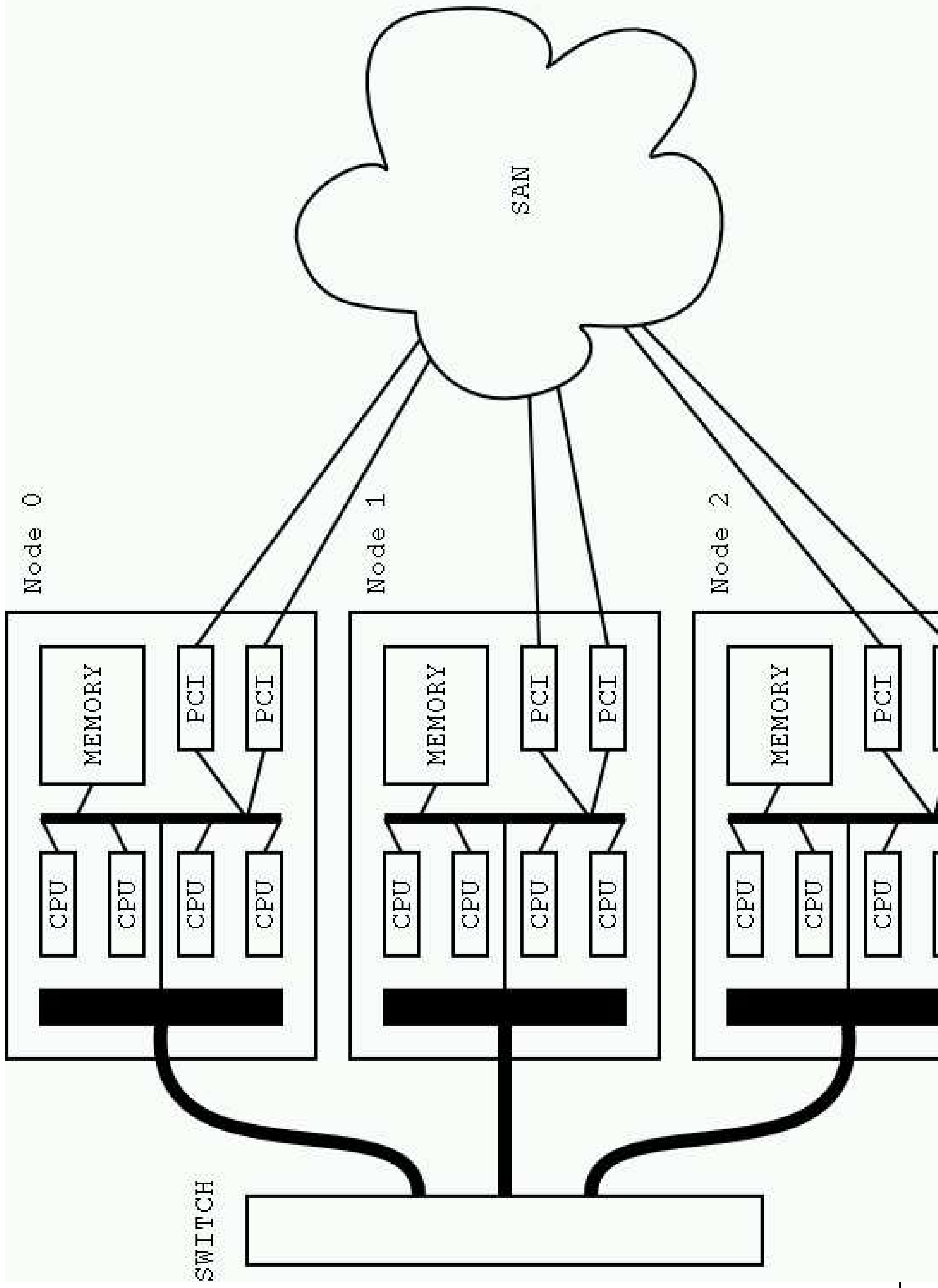
NUMA



Martin J. Bligh, Matt Dobson, Darren Hart

What is NUMA?

- Non-uniform memory architecture
- Different distances between CPUs, memory banks, IO.
- Local vs Remote
- NUMA ratios - and why they’re misleading
- "node" is a container.
- Trying mostly to acheive "locality"


SAN
Node 0
Node 1
Node 2
MEMORY
PCI
CPU
SWITCH

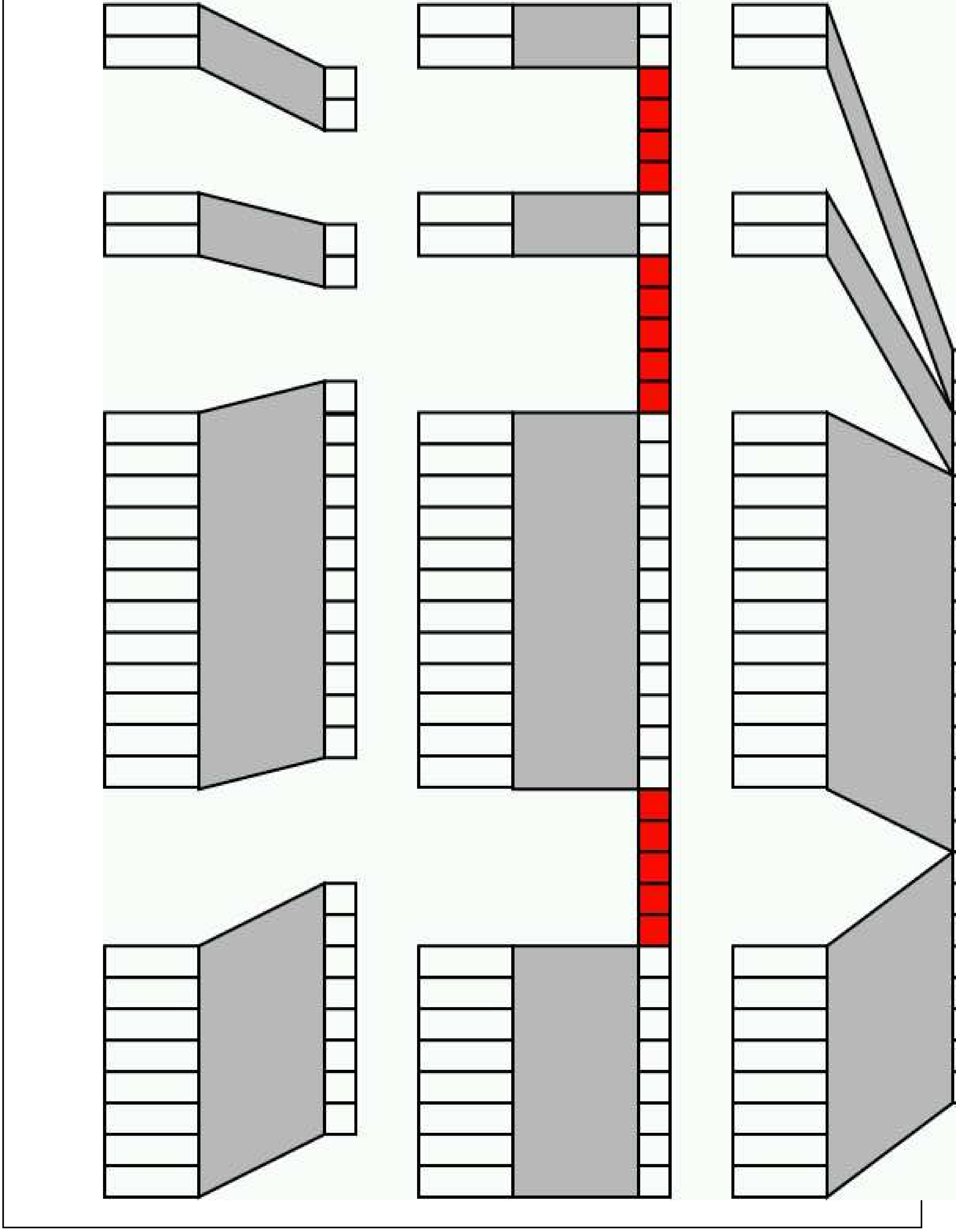
Why build a NUMA machine?

- Why not SMP?
- Faster local, not slower remote.
- What is the difference between NUMA and clusters?
- Why not use clusters? (SSI?)
- Why we mostly do things in the kernel, not in userspace.

Linux NUMA memory support

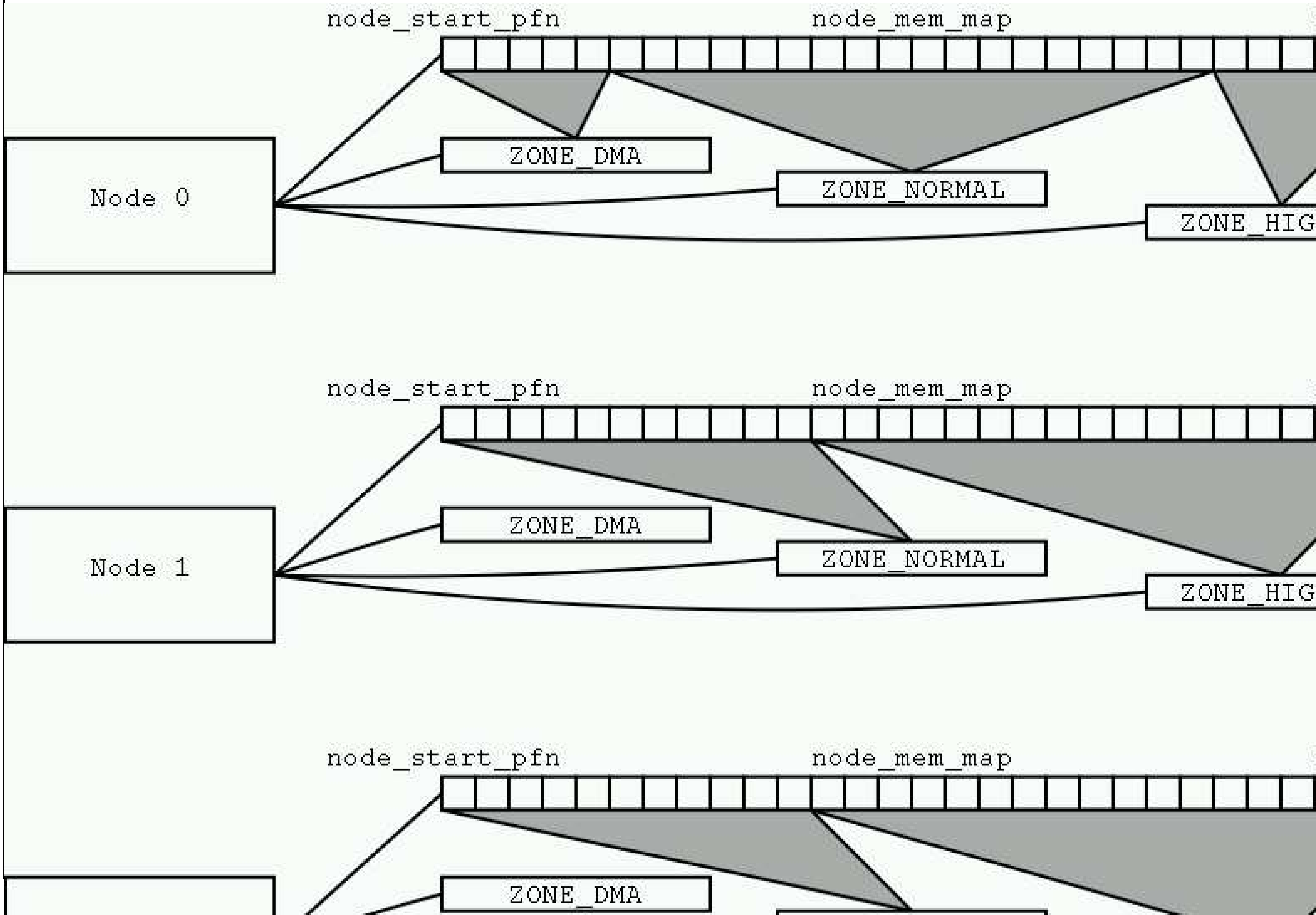
- (struct page) mem_map vs node_mem_map
- pg_data_t (struct node)

```
typedef struct pglist_data {
        struct zone node_zones[MAX_NR_ZONES];
        struct zonelist node_zonelists[MAX_NR_ZONES];
        int nr_zones;
        struct page *node_mem_map;
        struct bootmem_data *bdata;
        unsigned long node_start_pfn;
        unsigned long node_present_pages; /* total number of physical pages */
        unsigned long node_spanned_pages; /* total size of physical page range, including holes */
        int node_id;
        struct pglist_data *pgdat_next;
        wait_queue_head_t  kswapd_wait;
        struct task_struct *kswapd;
} pg_data_t;
```



Discontigmem and Nonlinear

- discontiguous memory
- CONFIG_NONLINEAR

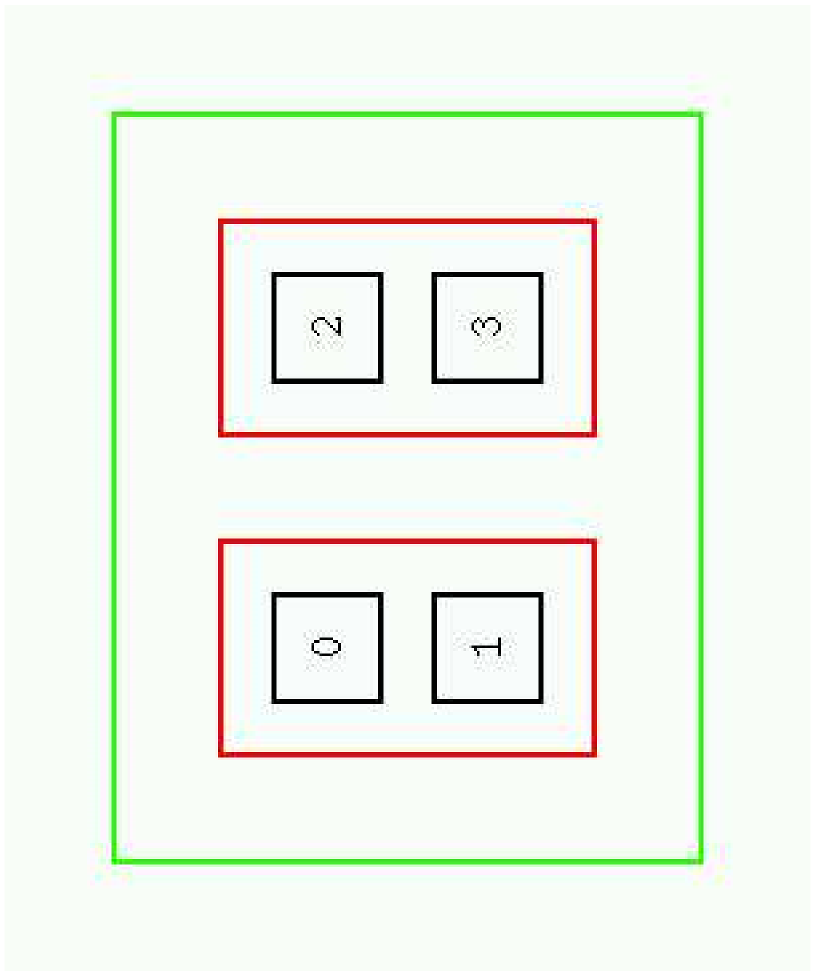

node_start_pfn
node_mem_map
Node 0
ZONE_DMA
ZONE_NORMAL
ZONE_HIG
node_start_pfn
node_mem_map
Node 1
ZONE_DMA
ZONE_NORMAL
ZONE_HIG
node_start_pfn
node_mem_map
ZONE_DMA

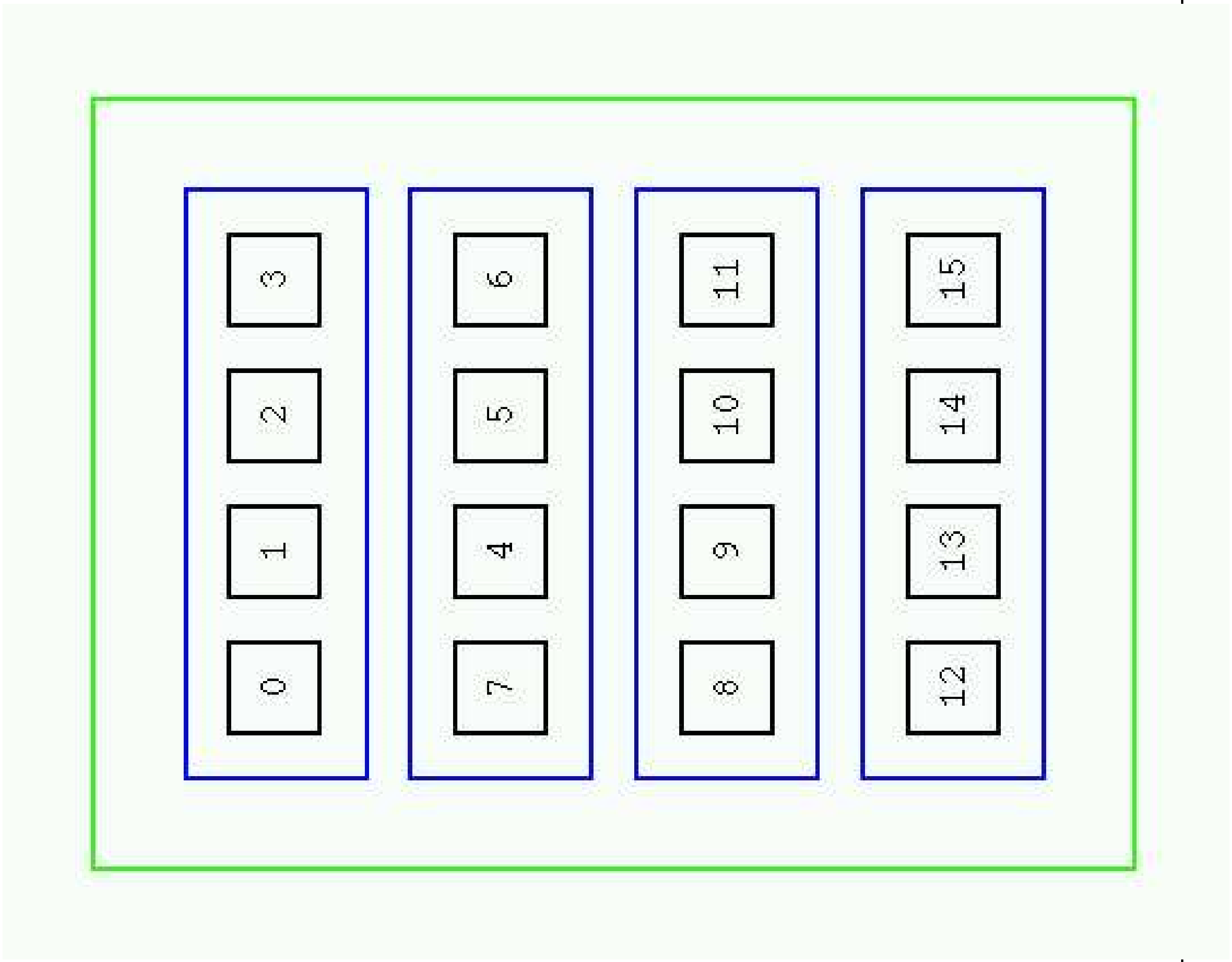
Using the NUMA memory support

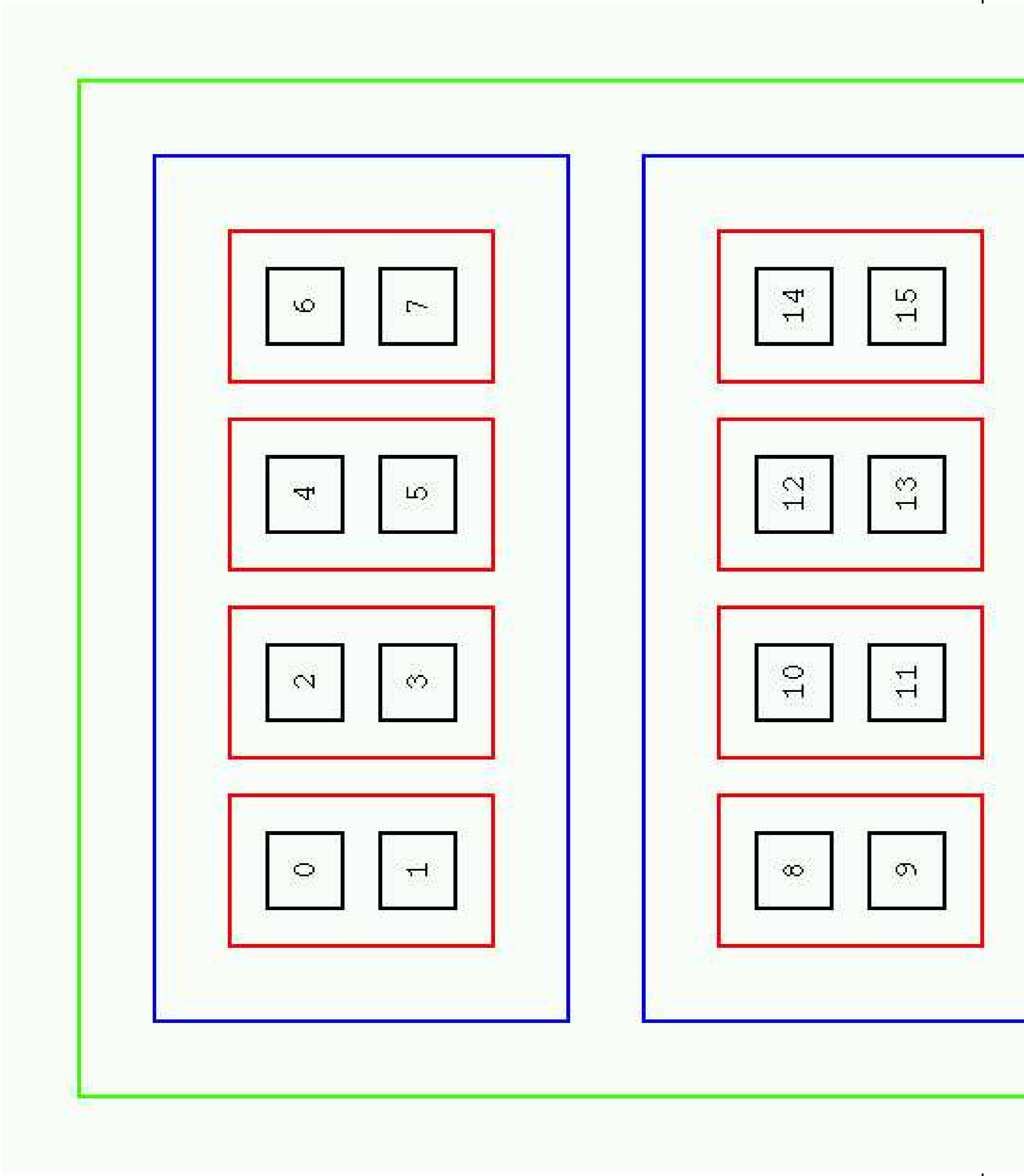
- Local allocation
- Replication - r/o, kernel, pagecache, other
- per-node LRU & locking

NUMA scheduler

- Why we need NUMA scheduler support (affinity, etc)
- First generation ... now moved to sched_domains
- sched_domains copes with more complex topologies.


2
3
0
1


0
1
2
3
7
4
5
6
8
9
10
11
12
13
14
15

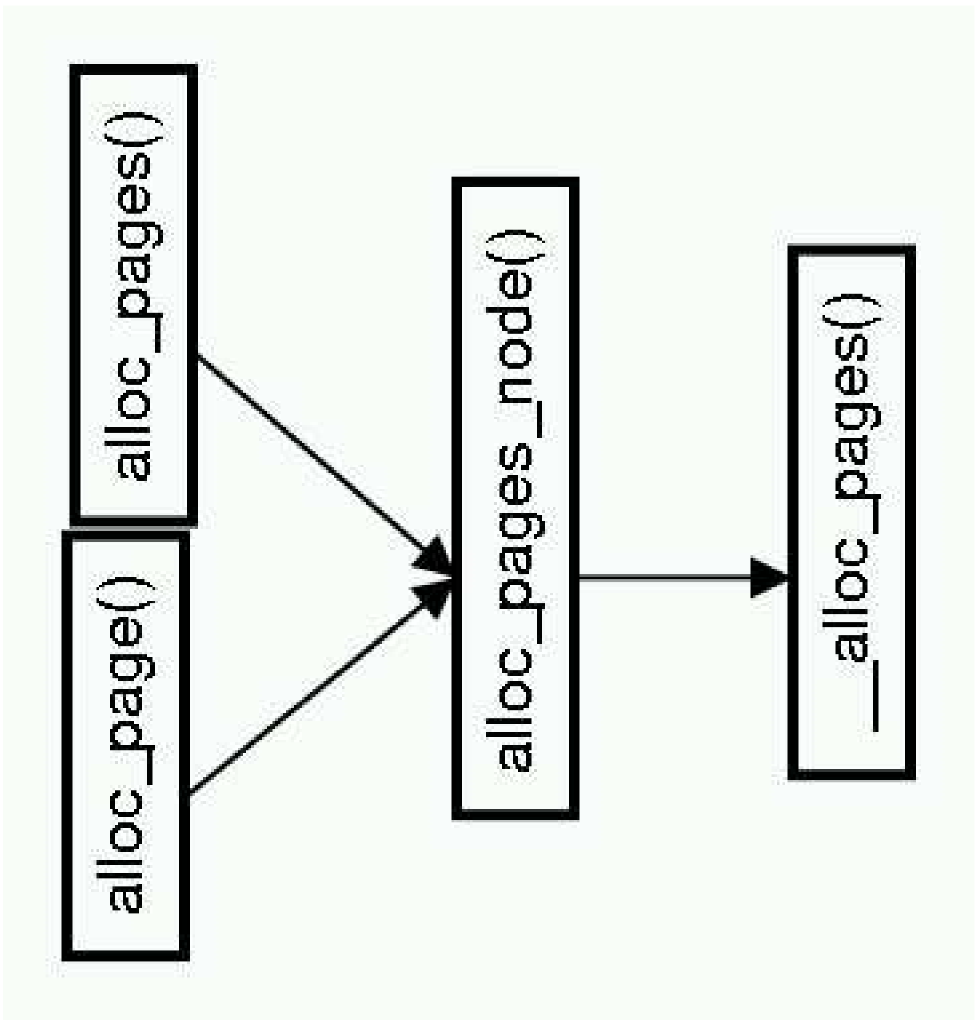

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

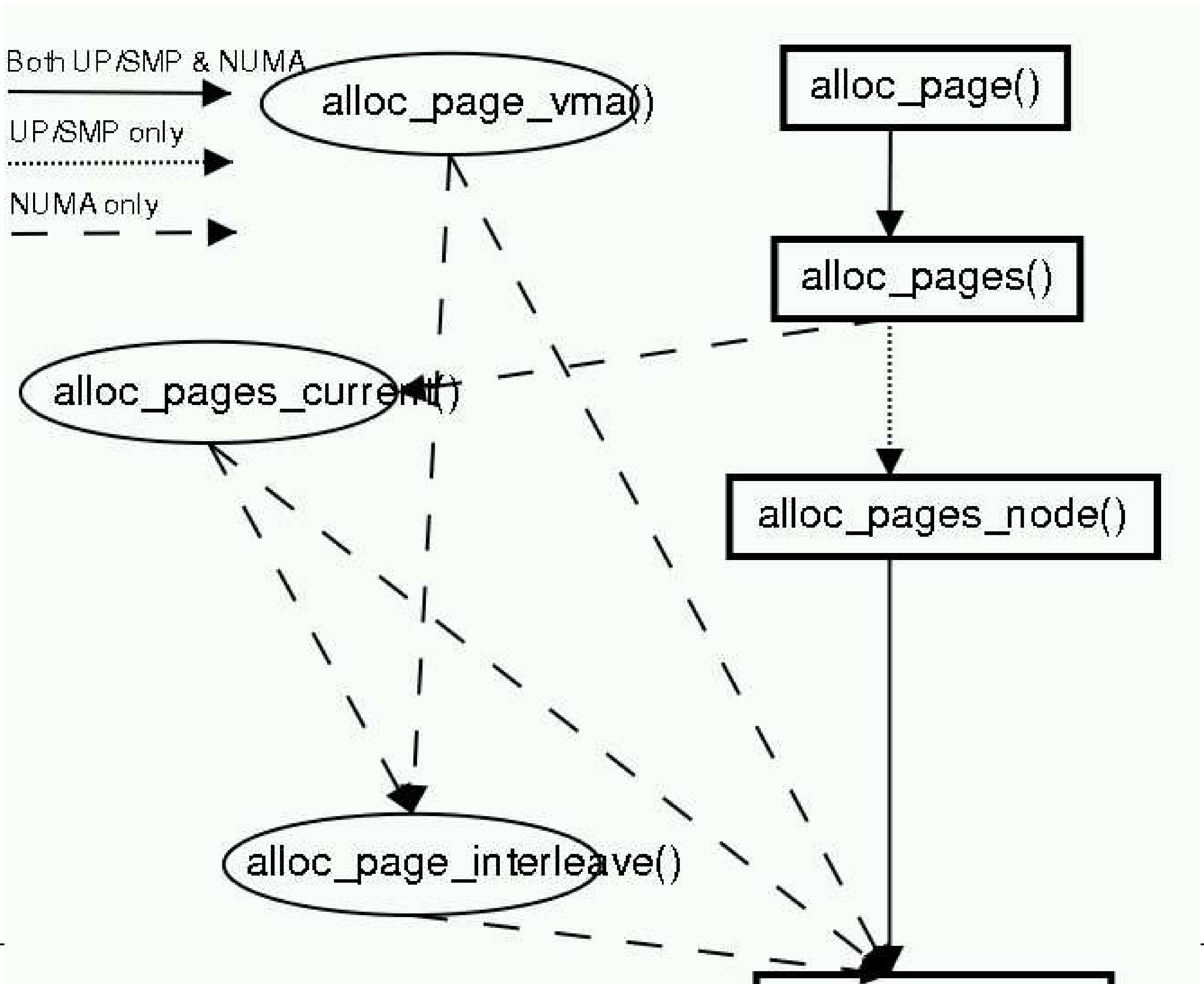
... more on sched_domains

- balance on exec / balance on clone
- event balancing vs active balancing
- parameters are abstracted, configurable

NUMA API (memory binding)

- Advantages and disadvantages.
- PREFERRED, BIND, INTERLEAVE, DEFAULT
- calls to set process or subregion of address space
- syscalls: sys_mbind, sys_set_mempolicy, sys_get_mempolicy.
- shared memory regions are dealt with via an rbtree
- Have discussed using anon_vma structures ... possibly.


alloc_pages()
alloc_page()
alloc_pages_node()
__alloc_pages()


Both UP/SMP & NUMA
UP/SMP only
NUMA only
alloc_page()
alloc_pages()
alloc_pages_node()
alloc_page_vma()
alloc_pages_current()
alloc_page_interleave()

To infinity, and beyond ....

- Better support for diversity of architectures
- Enhanced topology support
- Multipath IO
- NUMA-aware networking