vNUMA: Virtual Multiprocessors on Clusters of Workstations

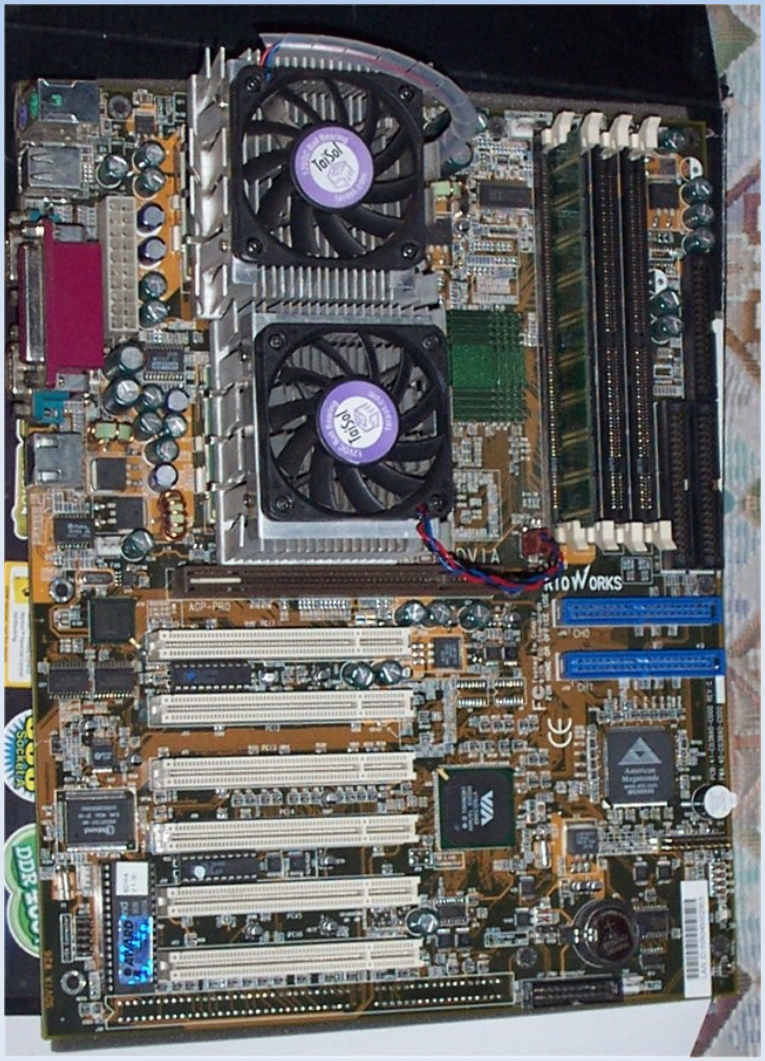
Matthew Chapman
University of New South Wales

matthewc@cse.unsw.edu.au

Parallel computing

Some problems can be solved faster with more than one processor.

Symmetric multiprocessors (SMP)

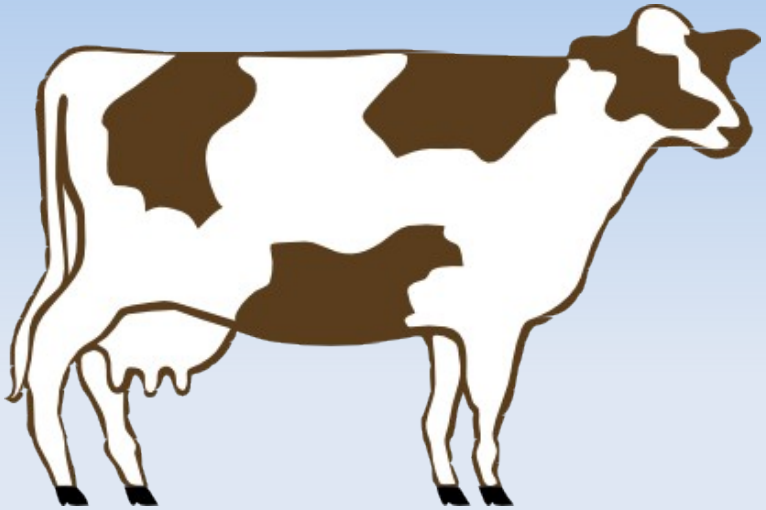


- ✔ Commodity
- ✔ Low communication overhead
- ✗ Limited # of CPUs

Large shared memory systems (NUMA)

- ✔ Natural extension of SMP
- ✔ Ease of use and programming
- ✗ Cost

Clusters of workstations (COW)



- ✔ Cost
- ✗ Complexity

Clusters of workstations (COW)

- ✔ Cost
- ✗ Complexity

Clusters of workstations (COW)

- ✔ Cost
- ✗ Complexity

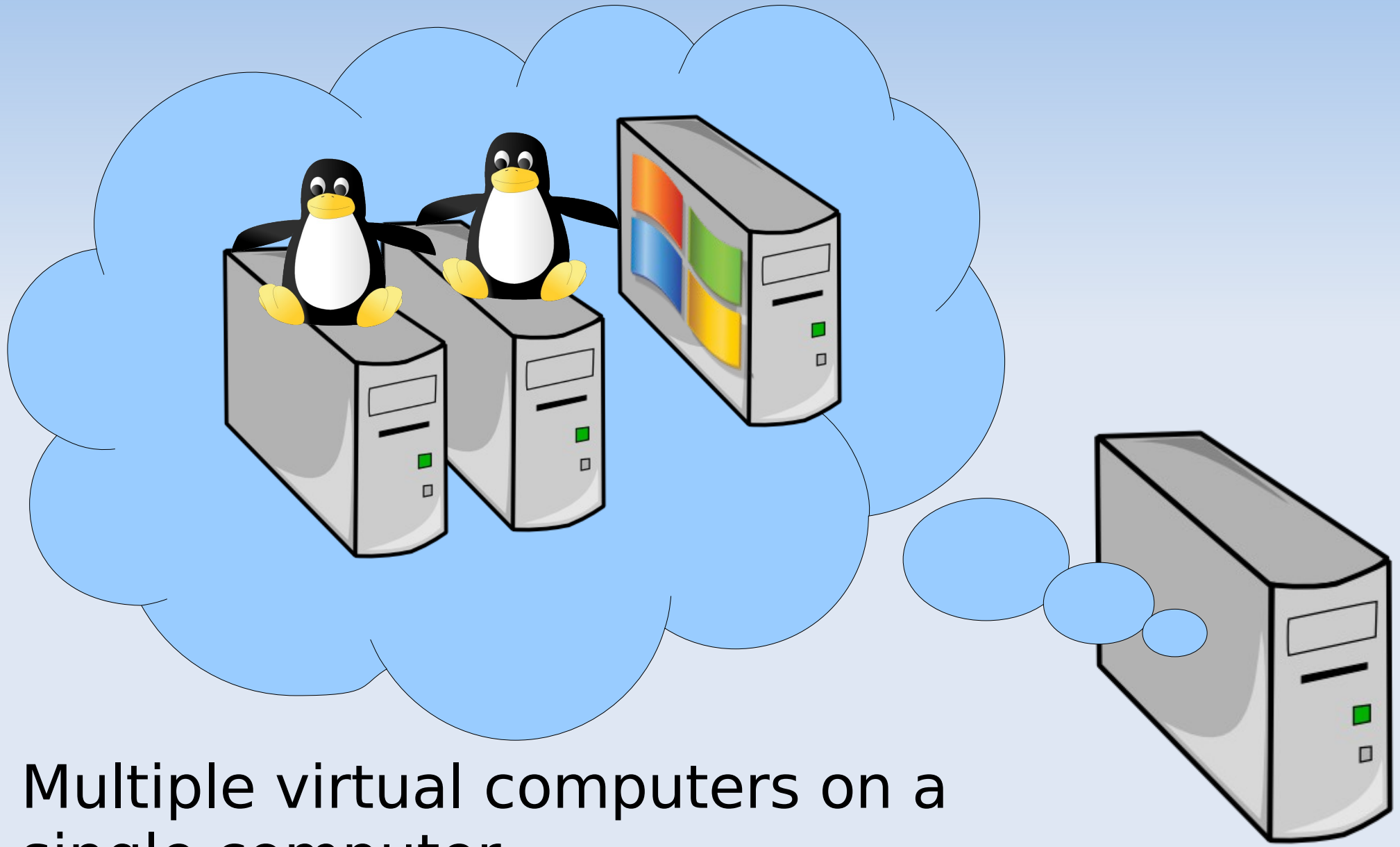
“If you were ploughing a field, which would you rather use? Two strong oxen or 1024 chickens?”

— Seymour Cray

Addressing the complexity

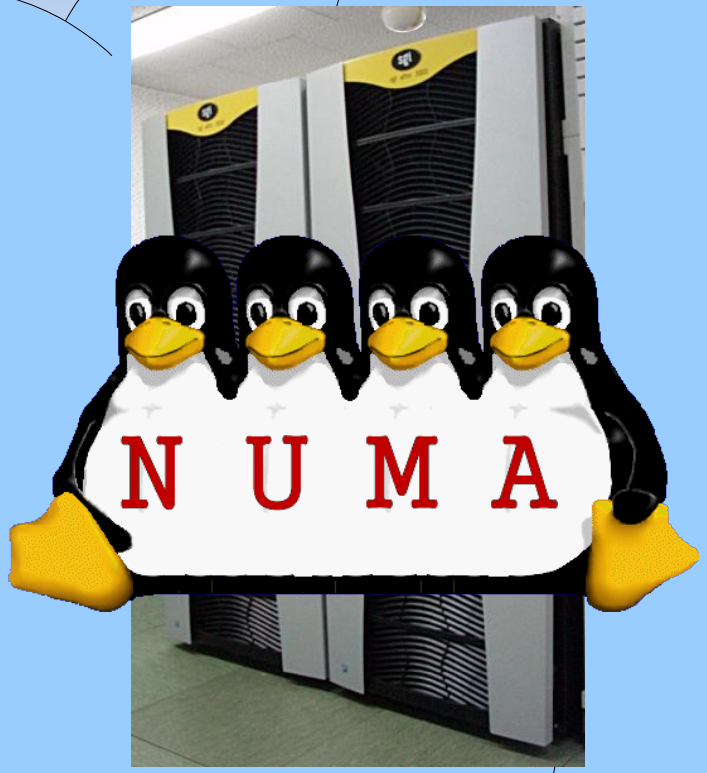
- Middleware
- Distributed operating systems
- Virtualization?

Virtualization



Multiple virtual computers on a single computer

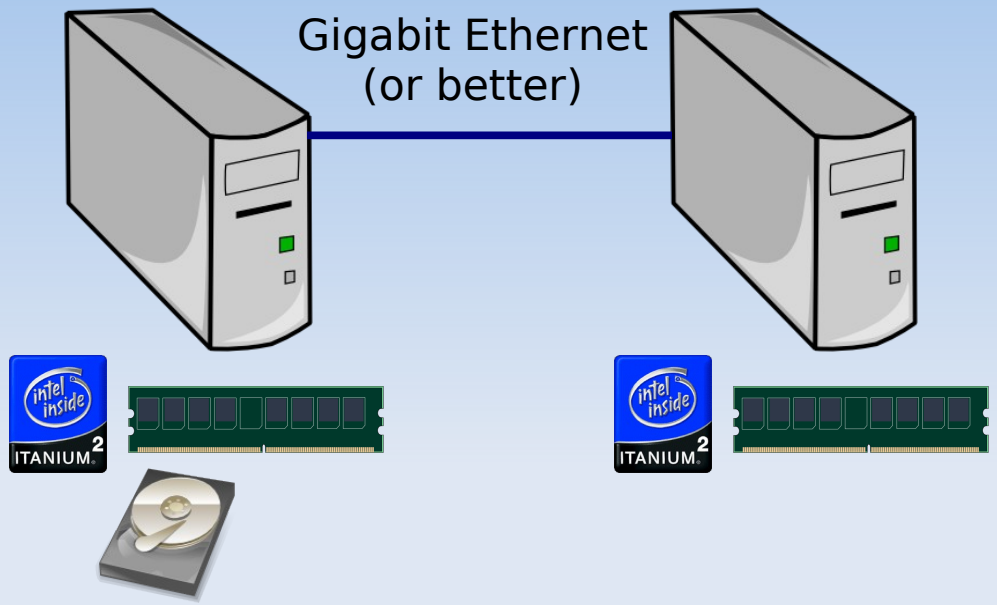
Virtualization



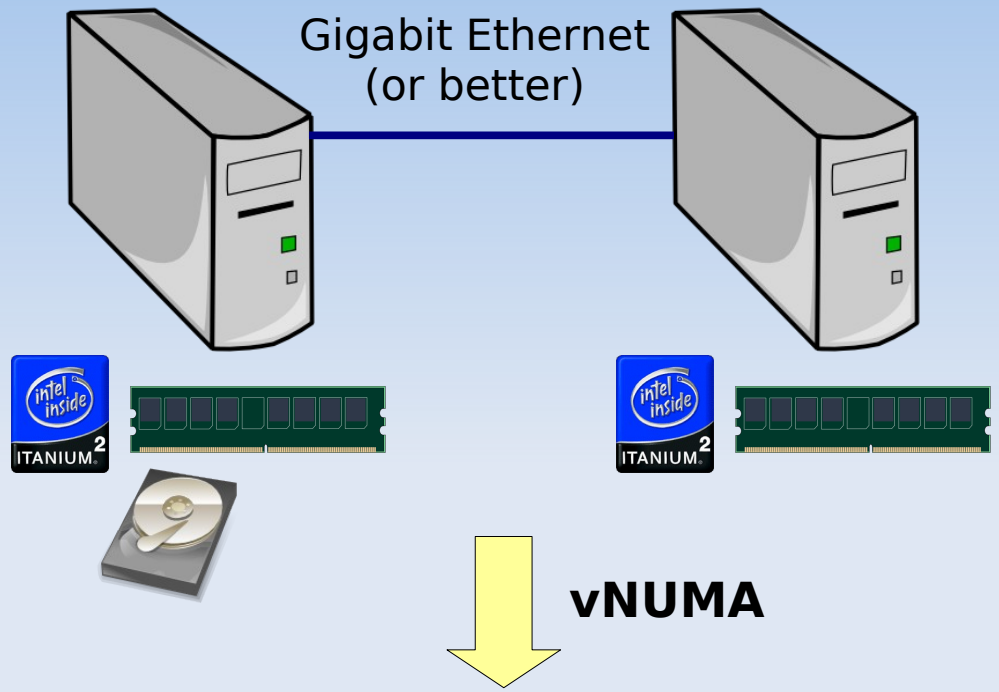
Virtual NUMA system
on a cluster

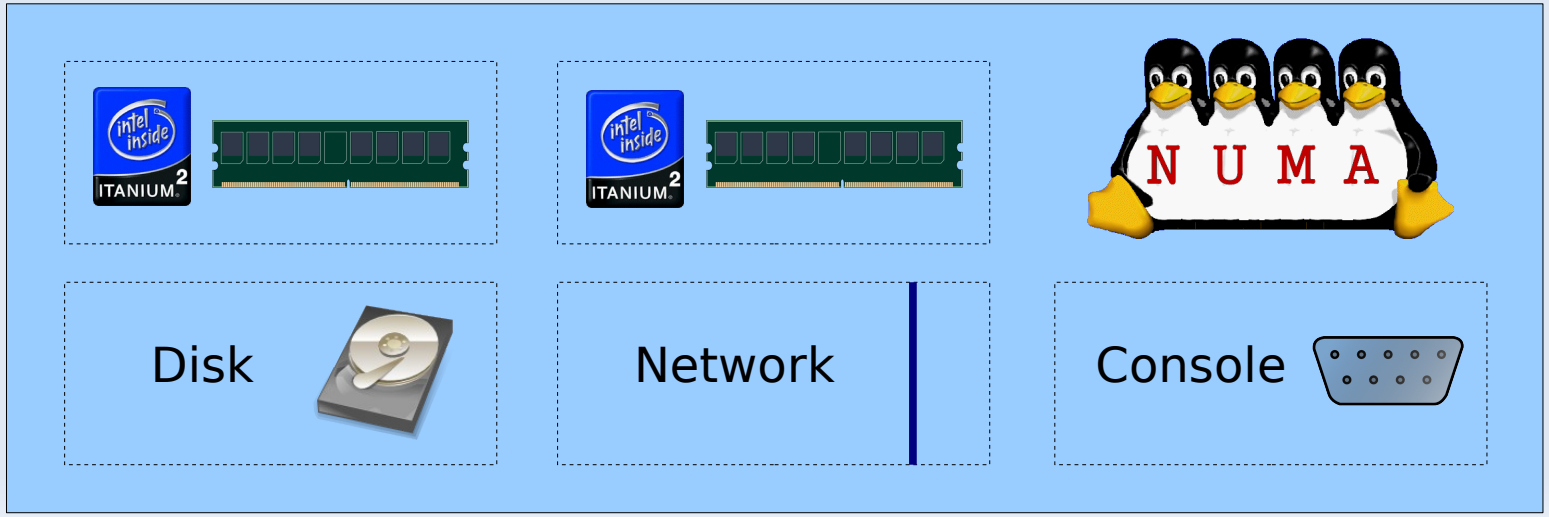
?

vNUMA


Gigabit Ethernet
(or better)
intel inside
ITANIUM 2
intel inside
ITANIUM 2

vNUMA


Gigabit Ethernet
(or better)
vNUMA


N U M A
Disk
Network
Console

vNUMA

- Thin hypervisor
- Booted from ELILO
- Discovers other resources on network

Distributed Shared Memory (DSM)

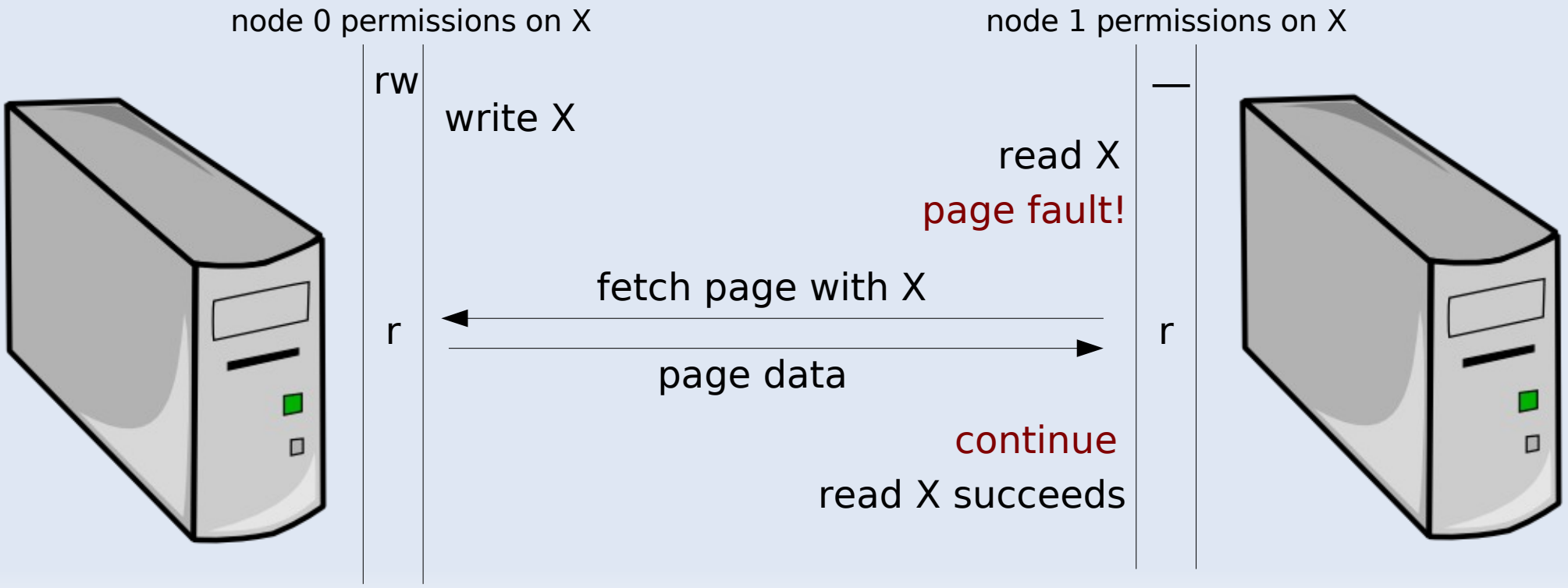
- Simulate shared memory using virtual memory mechanisms
- Page granularity (4KB)

Distributed Shared Memory (DSM)

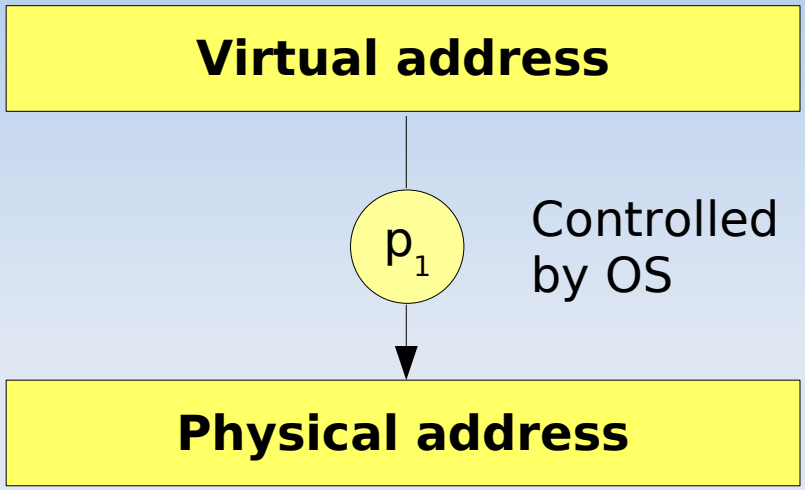
- Simulate shared memory using virtual memory mechanisms
- Page granularity (4KB)

Distributed Shared Memory (DSM)

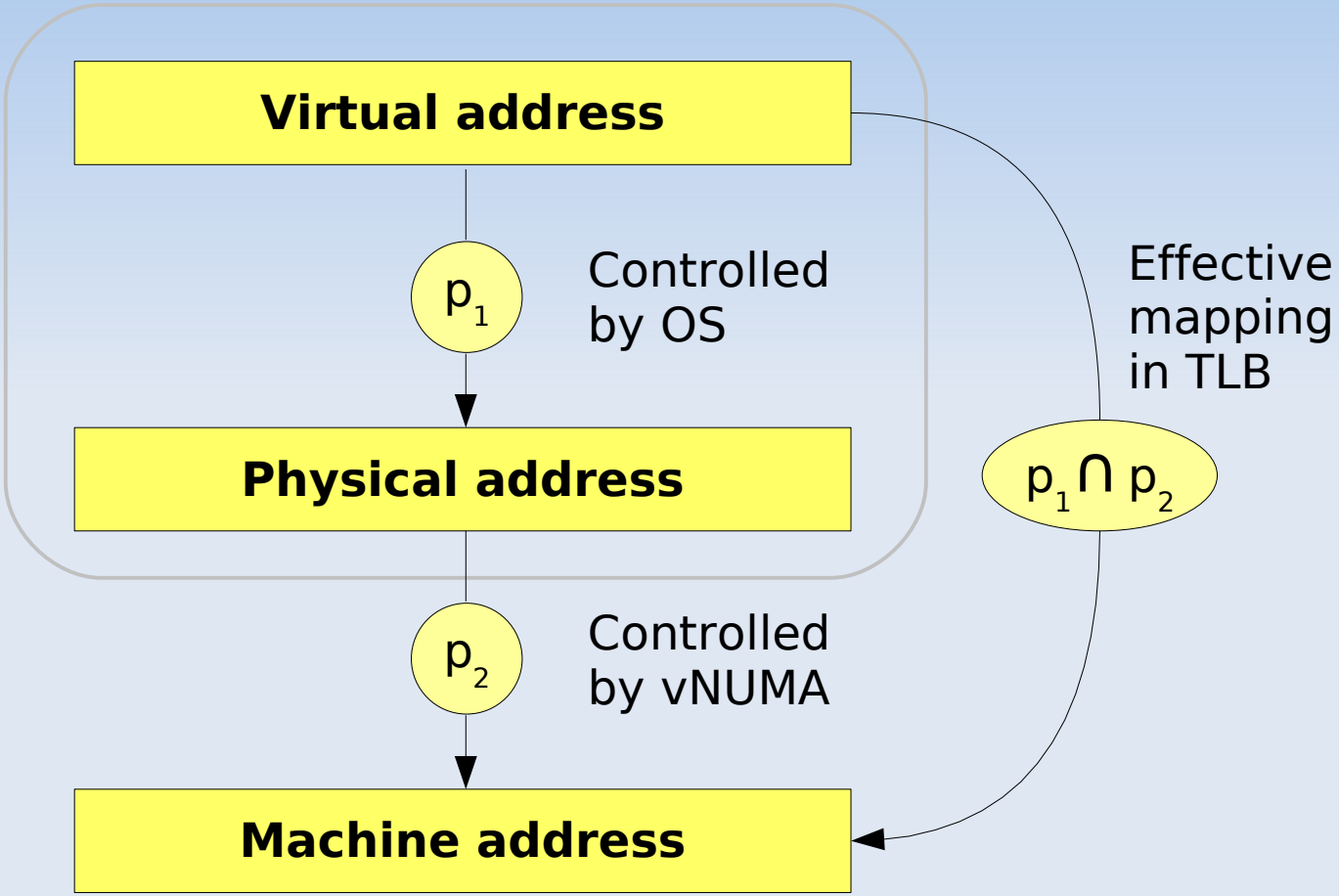
- Simulate shared memory using virtual memory mechanisms
- Page granularity (4KB)



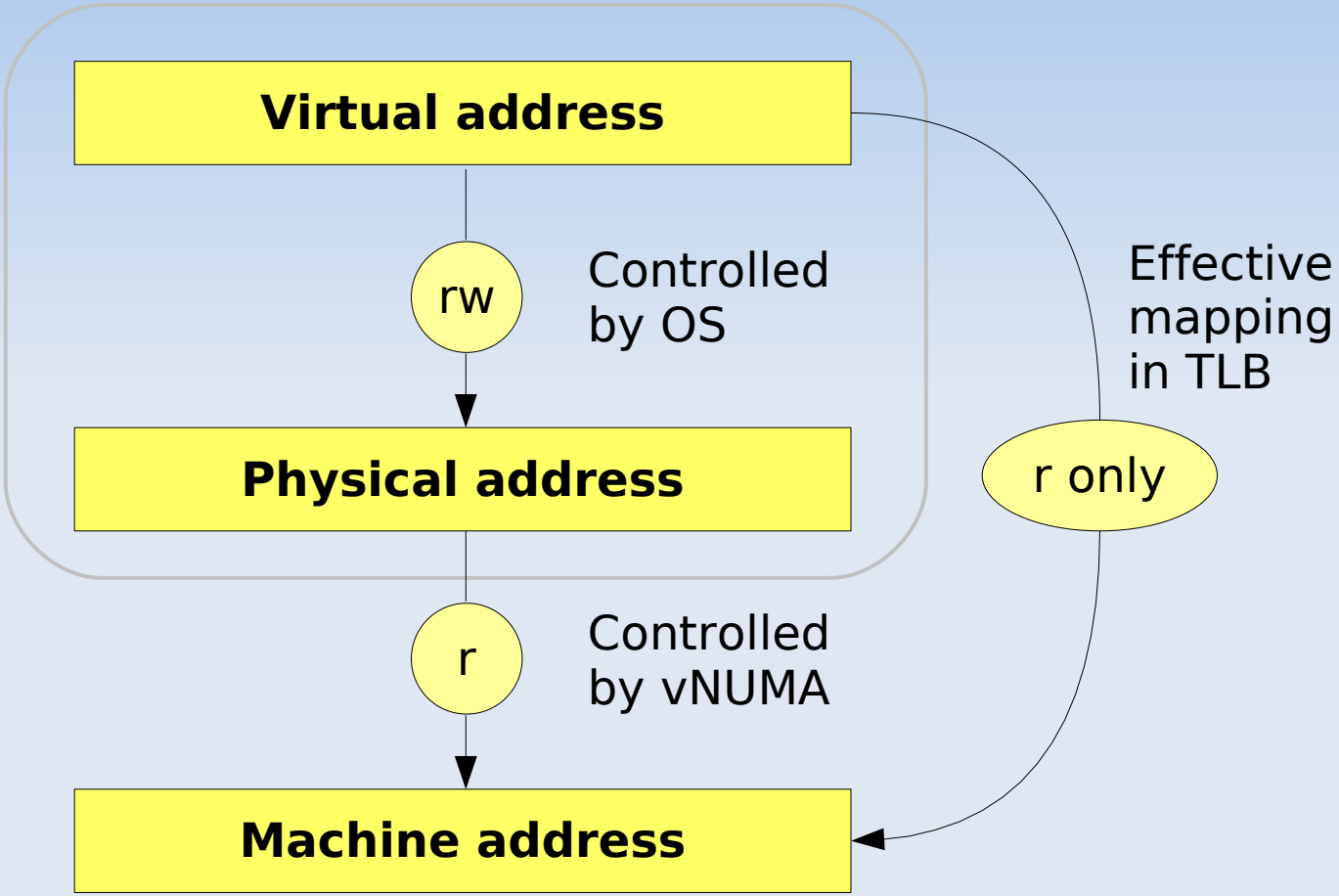
Memory management in an OS



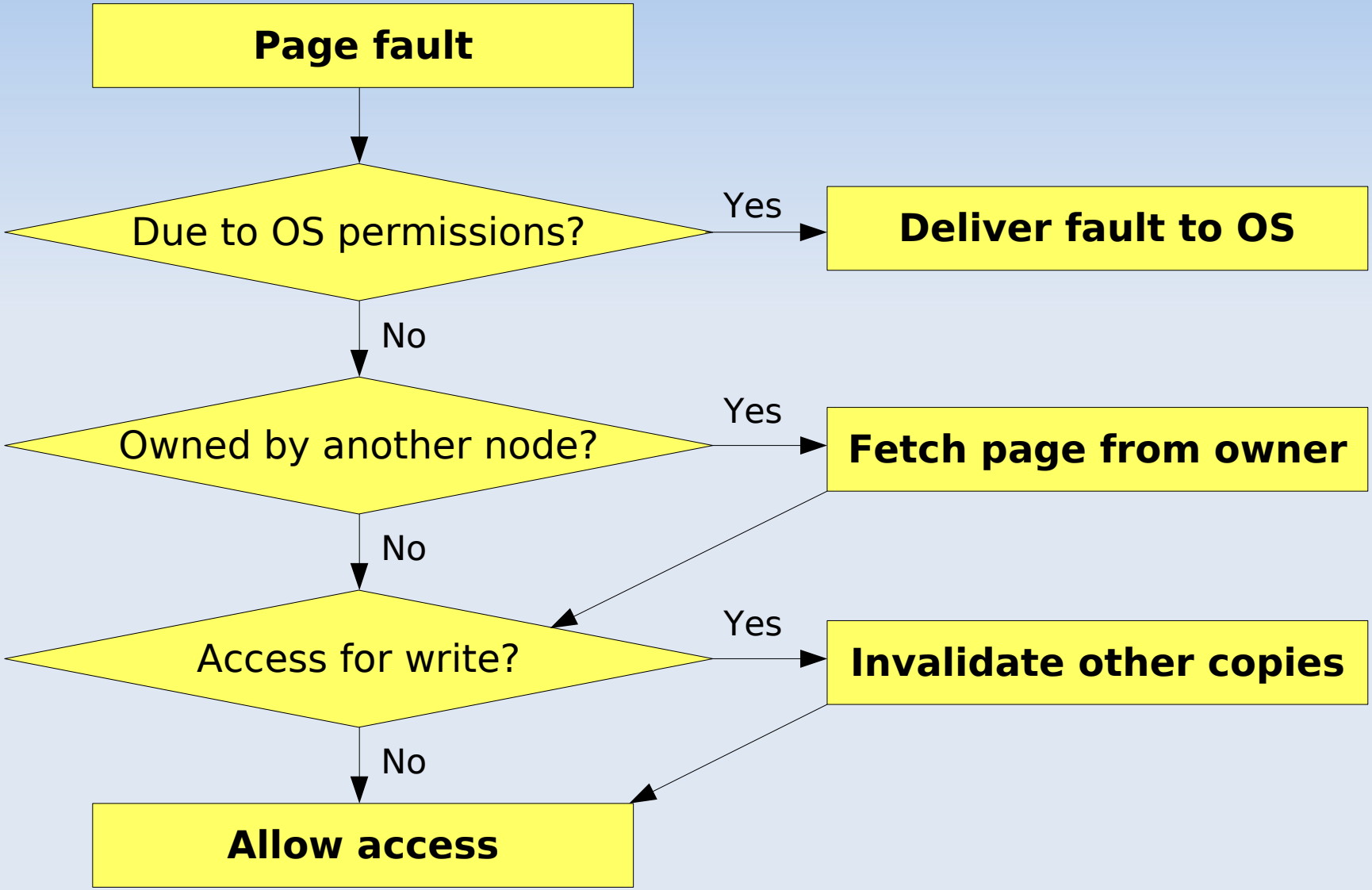
Memory management in vNUMA



Memory management in vNUMA

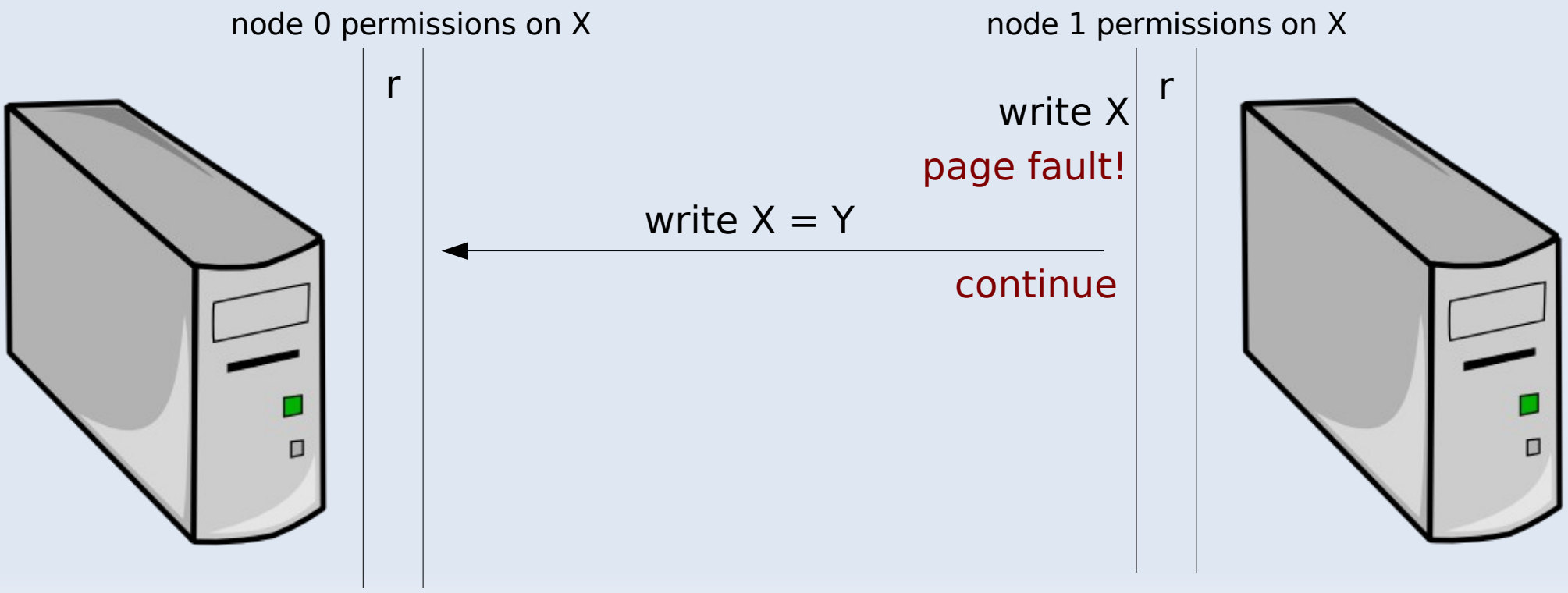


Page fault handling (simplified)



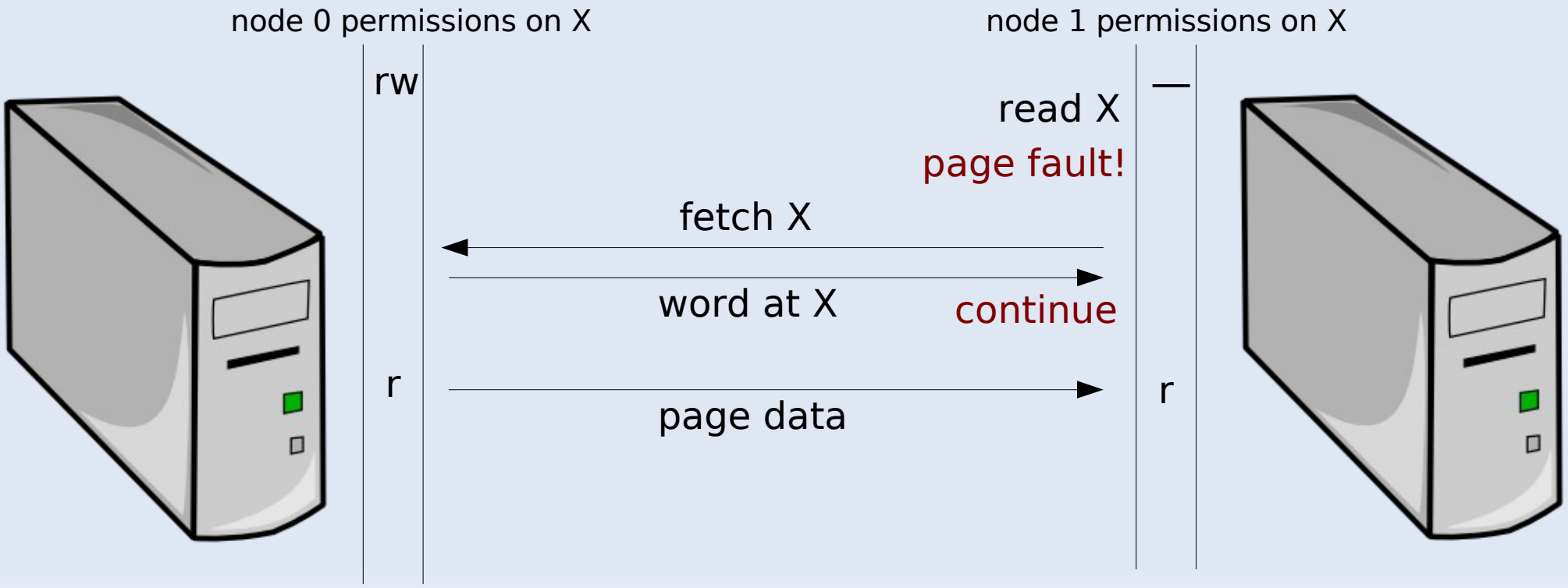
Optimisation: Write updates

- Transmit individual writes instead of page-level invaludations in some cases
- Writes queued and sent in batches

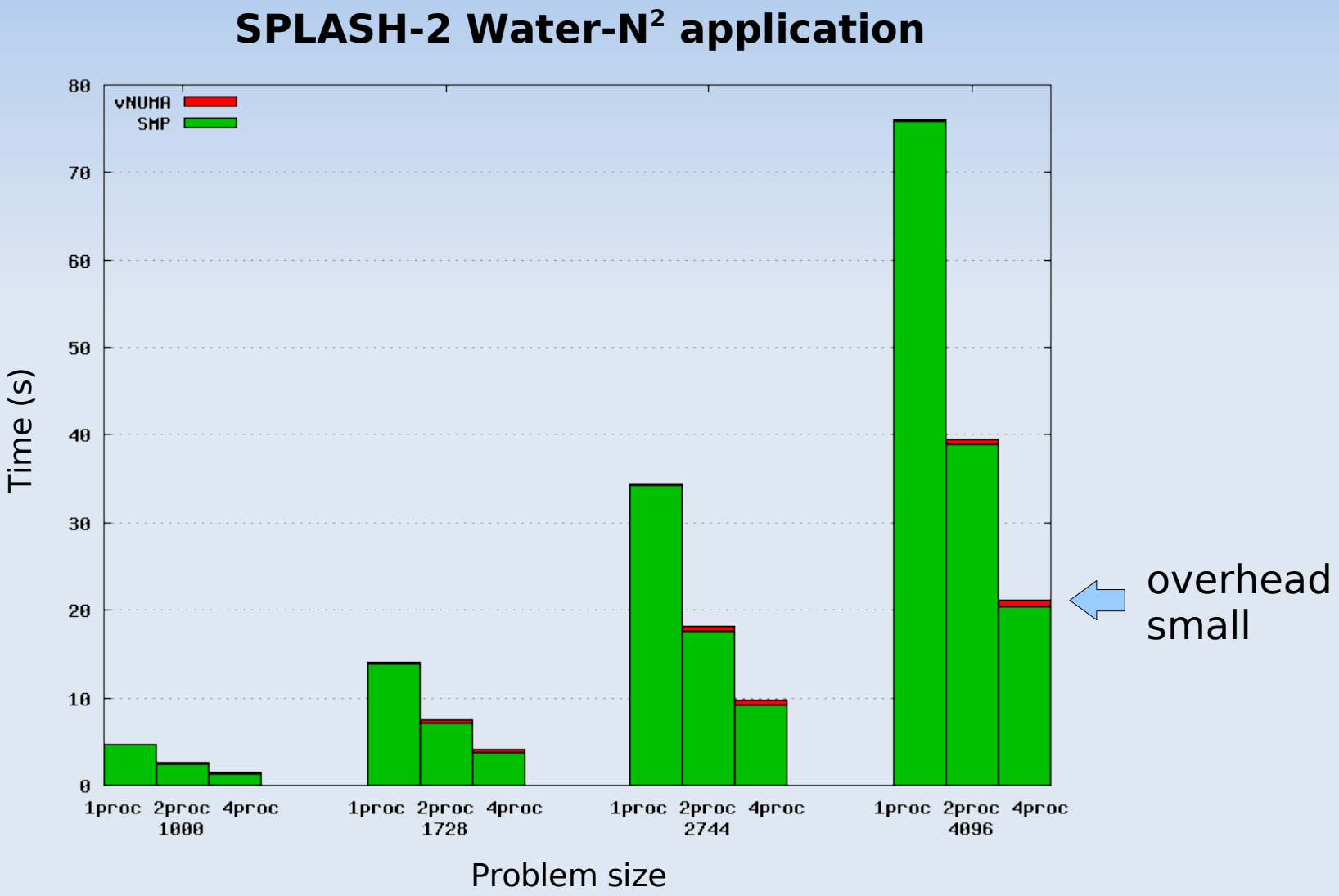


Optimisation: Critical Word First

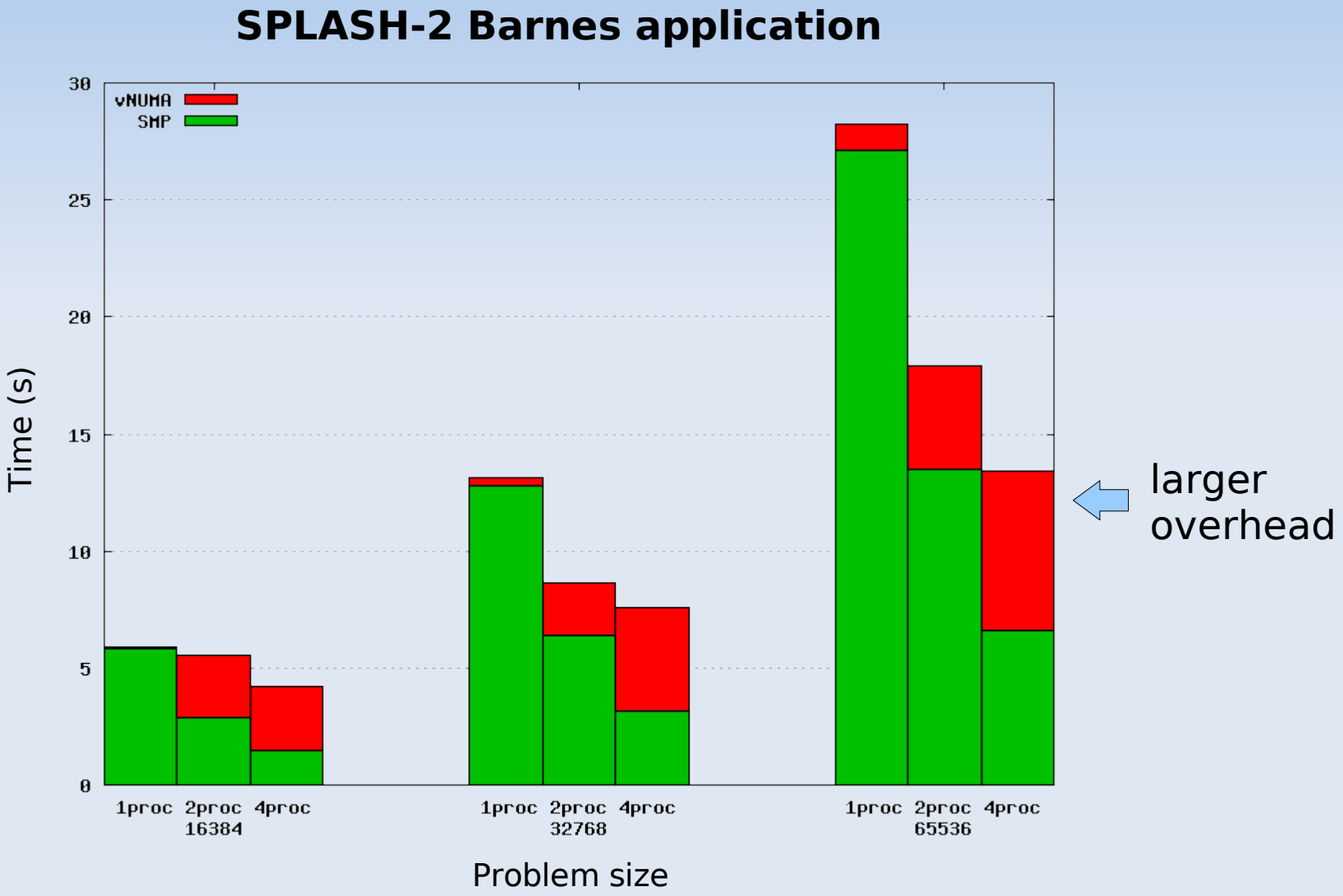
- Remote reads return requested word first
- Allows faster restart



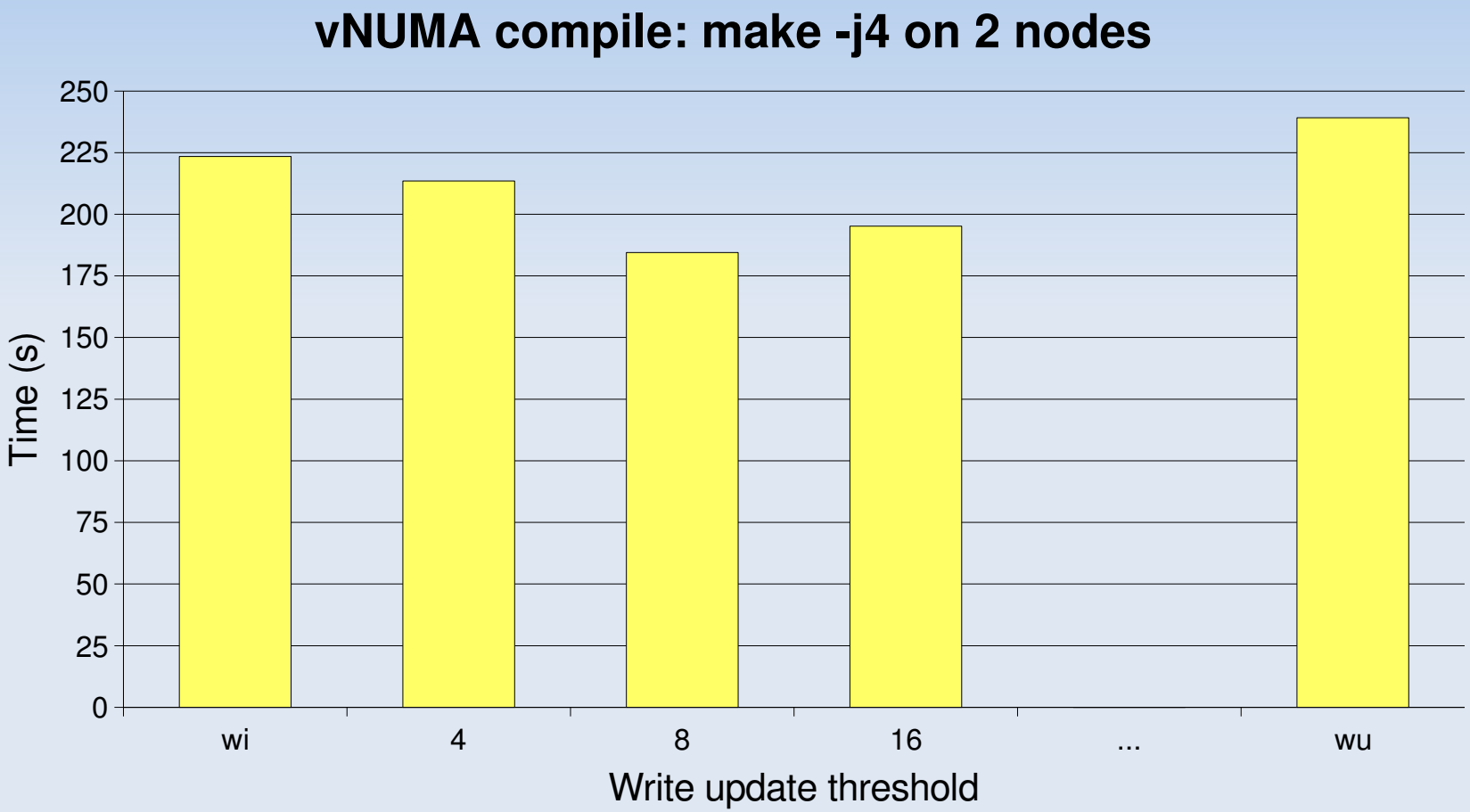
Performance: Scientific app.



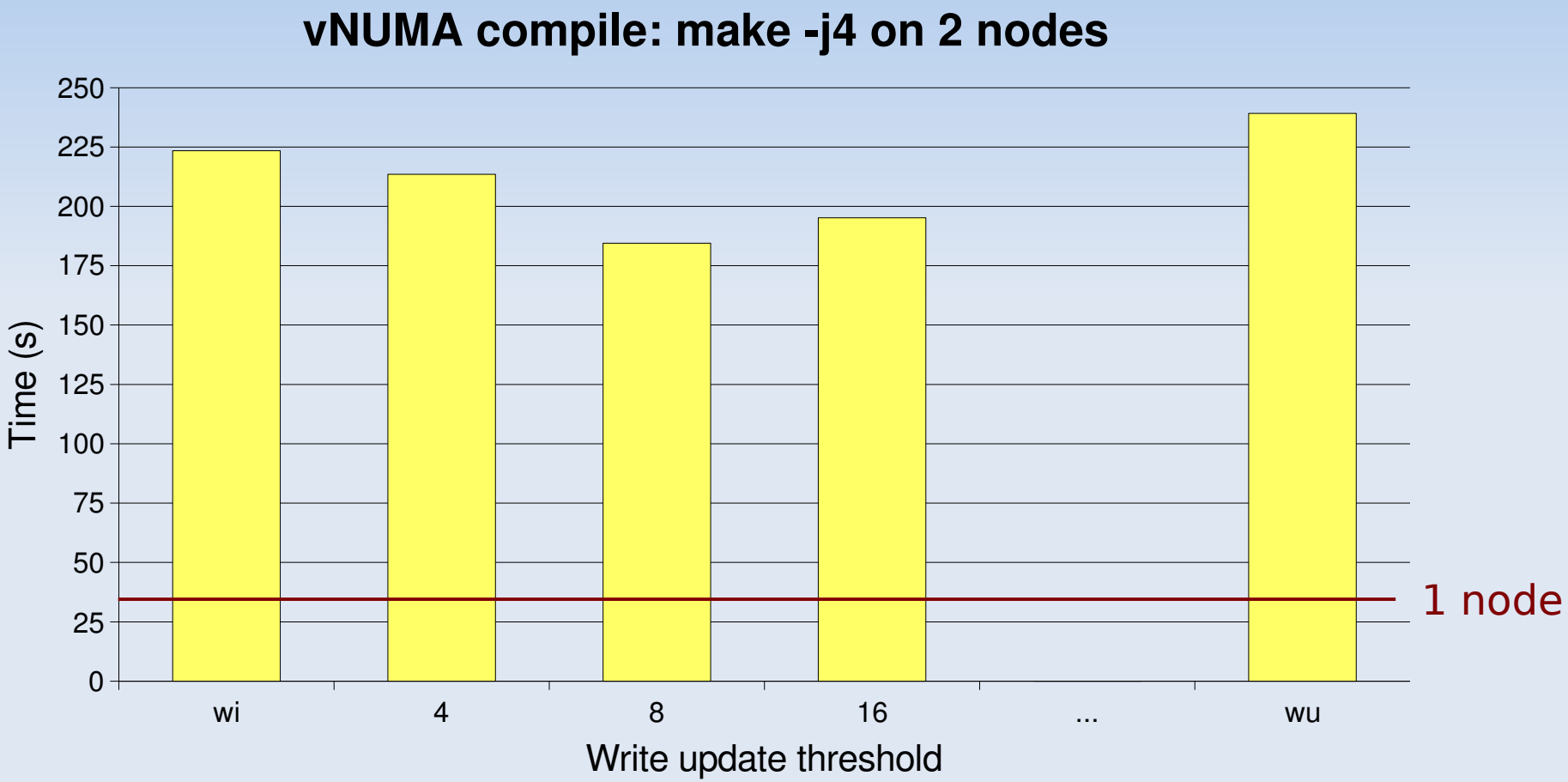
Performance: Scientific app.



Performance: Parallel compile

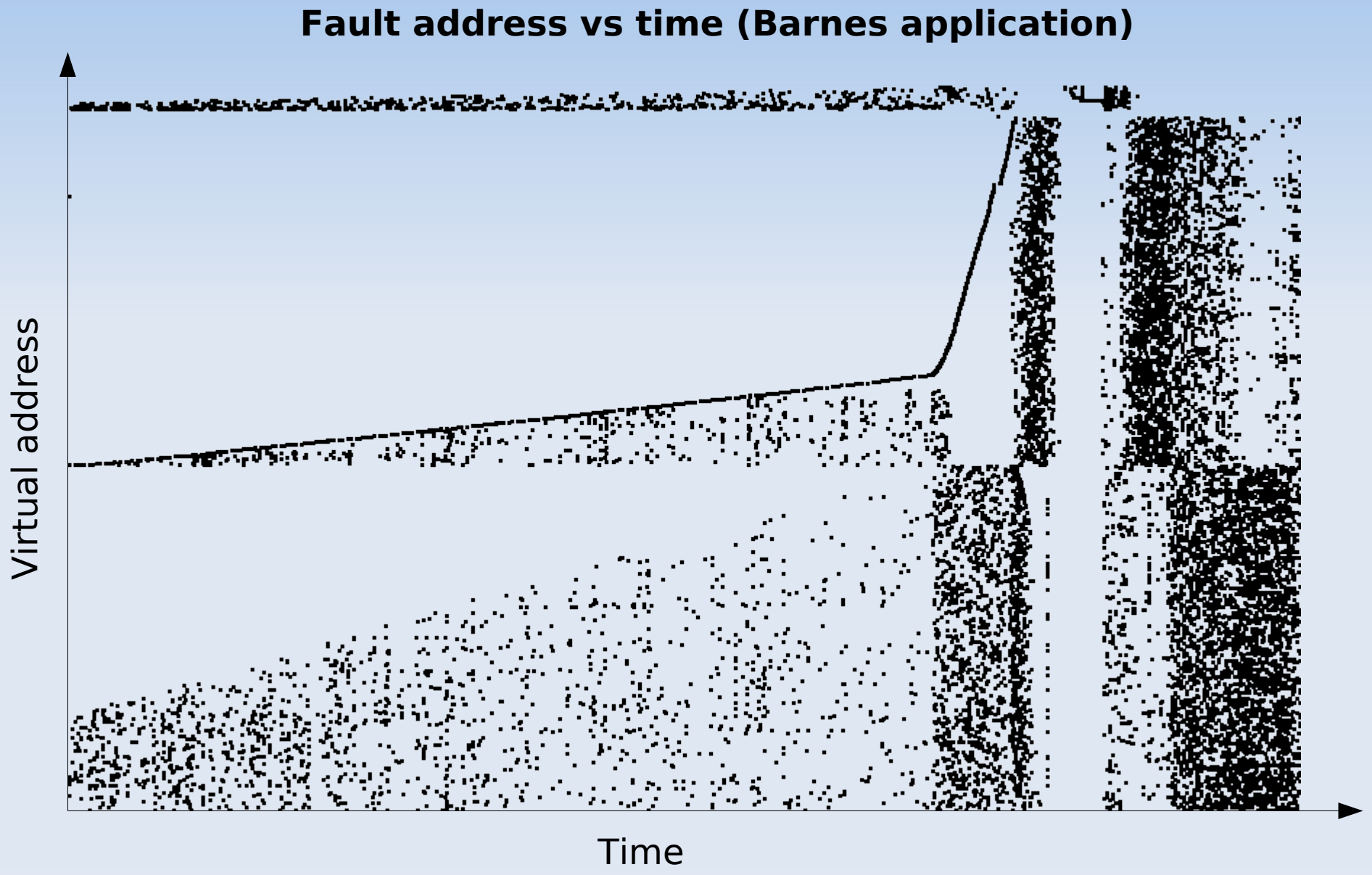


Performance: Parallel compile



Severe negative scaling!

Profiling tools: vnumaviz



Profiling tools: vnumaprof

Compile benchmark profile

```
Total DSM overhead: 280396.830 ms in 211417r/1825x/135040w/106995i/371367s
              User: 16231.443 ms in 22677r/1823x/8599w/5793i/287s (5.7%)
            Kernel: 264165.387 ms in 188740r/2x/126441w/101202i/371080s
(94.2%)
              Read: 106757.495 ms in 211417r/0x/0w/0i/0s (38.1%)
           Execute: 828.677 ms in 0r/1825x/0w/0i/0s (0.3%)
             Write: 44071.038 ms in 0r/0x/135040w/0i/0s (15.7%)
        Invalidate: 31117.793 ms in 0r/0x/0w/106995i/0s (11.1%)
         Semaphore: 97621.827 ms in 0r/0x/0w/0i/371367s (34.8%)

 @ 0xa0000001004d19a0: 30362.010 ms in 0r/0x/13684w/8060i/88400s
  in __raw_spin_lock_flags() at include/asm/spinlock.h:78
 @ 0xa0000001002eae30: 7889.798 ms in 0r/0x/17774w/4268i/0s
  in clear_page()
 @ 0xa000000100100d10: 6726.483 ms in 0r/0x/2091w/2024i/22259s
  in atomic_add_negative() at include/asm/atomic.h:135
 @ 0xa0000001002dd940: 6315.203 ms in 0r/0x/685w/916i/20690s
  in _atomic_dec_and_lock() at lib/dec_and_lock.c:24
 @ 0xa0000001000ce370: 5359.691 ms in 0r/0x/2582w/2605i/17701s
  in find_get_page() at include/linux/mm.h:332
 @ 0xa0000001004d2a90: 4411.197 ms in 0r/0x/1484w/924i/12376s
  in lock_kernel() at include/asm/spinlock.h:78
```

Improving performance

- Optimised locks (sub-page granularity)
- Better I/O distribution
- More virtual CPUs than physical CPUs (multithreading)
- Faster interconnect (Infiniband?)

Future possibilities

- SMP support
- Other architectures
- Fault tolerance

Questions?