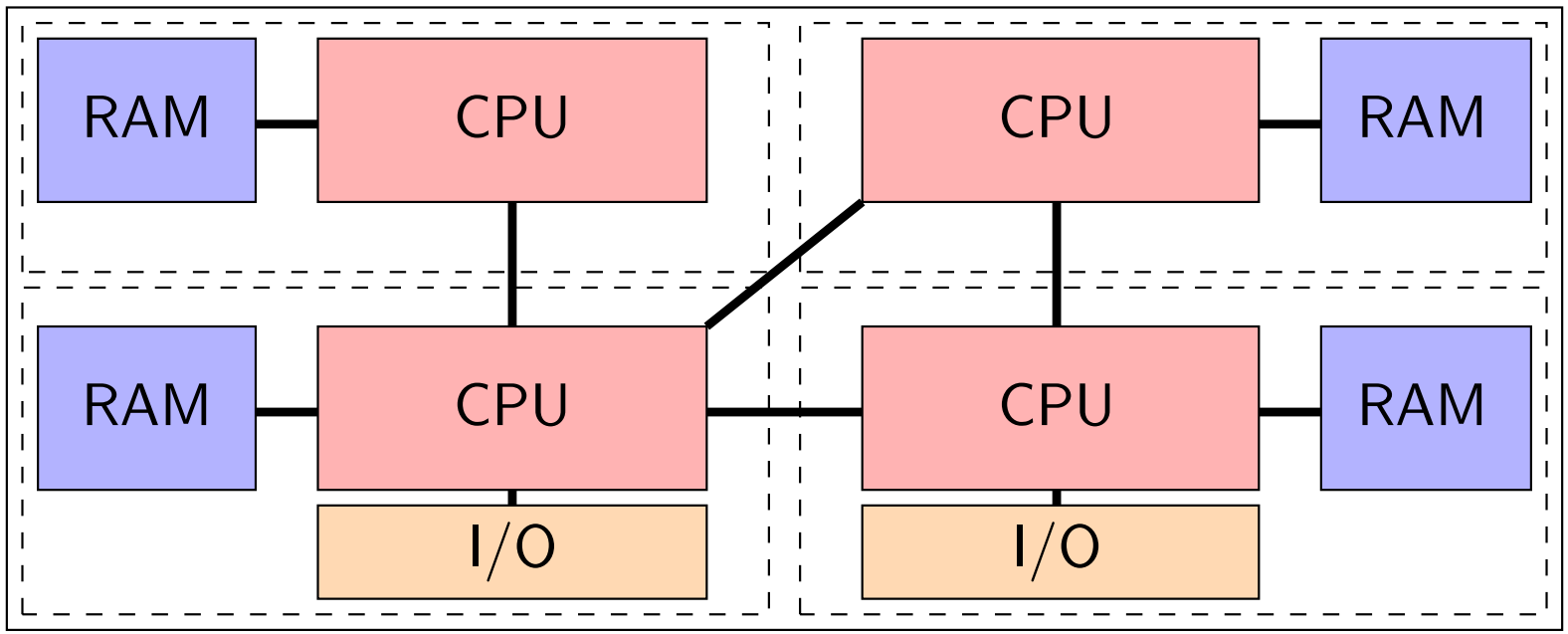
An interface to implement NUMA policies in the Xen hypervisor

Gauthier Voron — INRIA/LIP6
Gaël Thomas — Telecom SudParis
Vivien Quéma — Grenoble INP / ENSIMAG
Pierre Sens — INRIA/LIP6

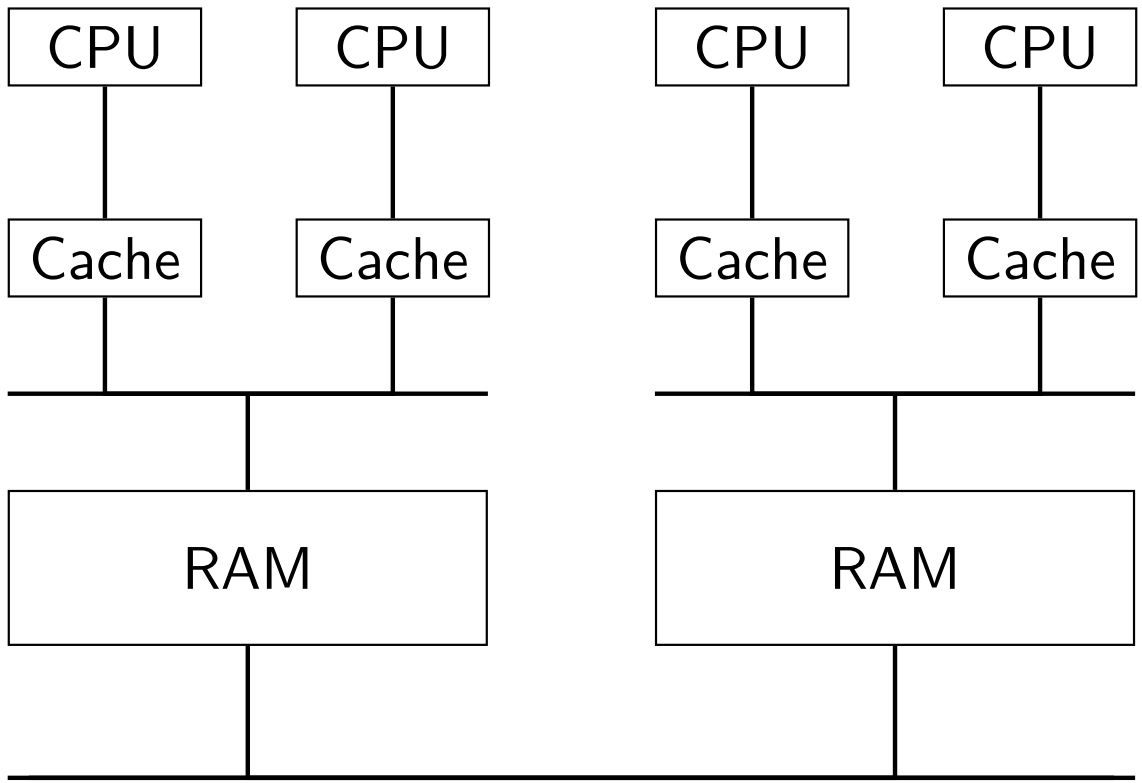
Complex computations require NUMA

- Modern applications need computing power
- NUMA architectures provide computing power but are complex to use



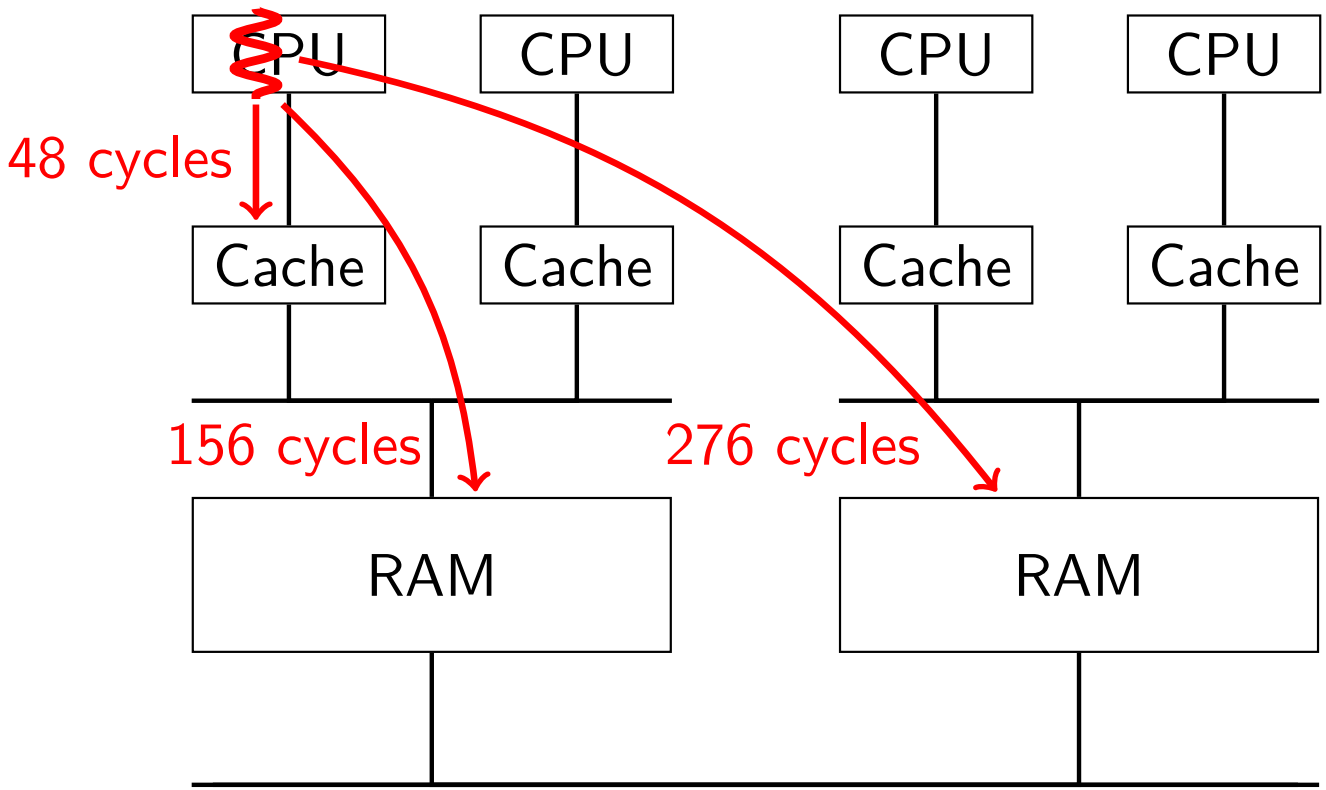
NUMA performance criteria

- Accessing data from local memory
- Avoiding to create contention



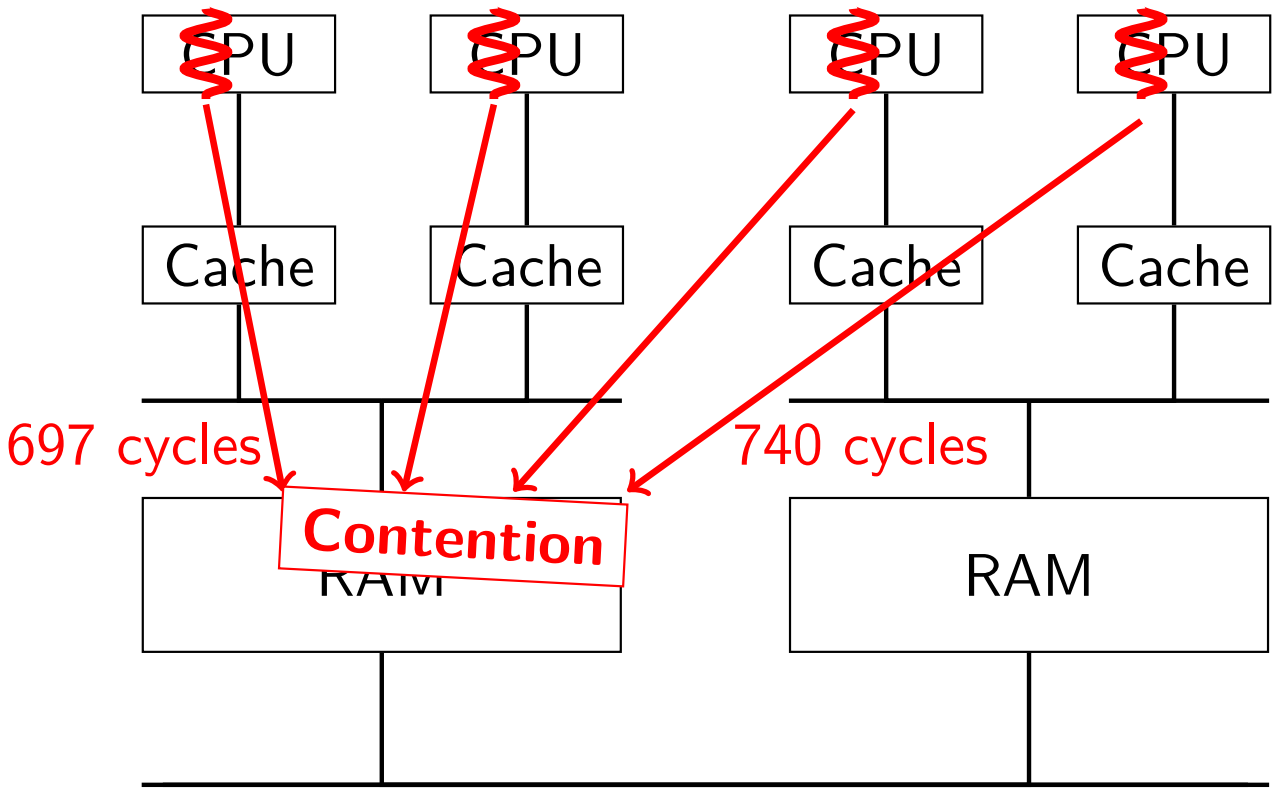
NUMA performance criteria

- Accessing data from local memory
- Avoiding to create contention



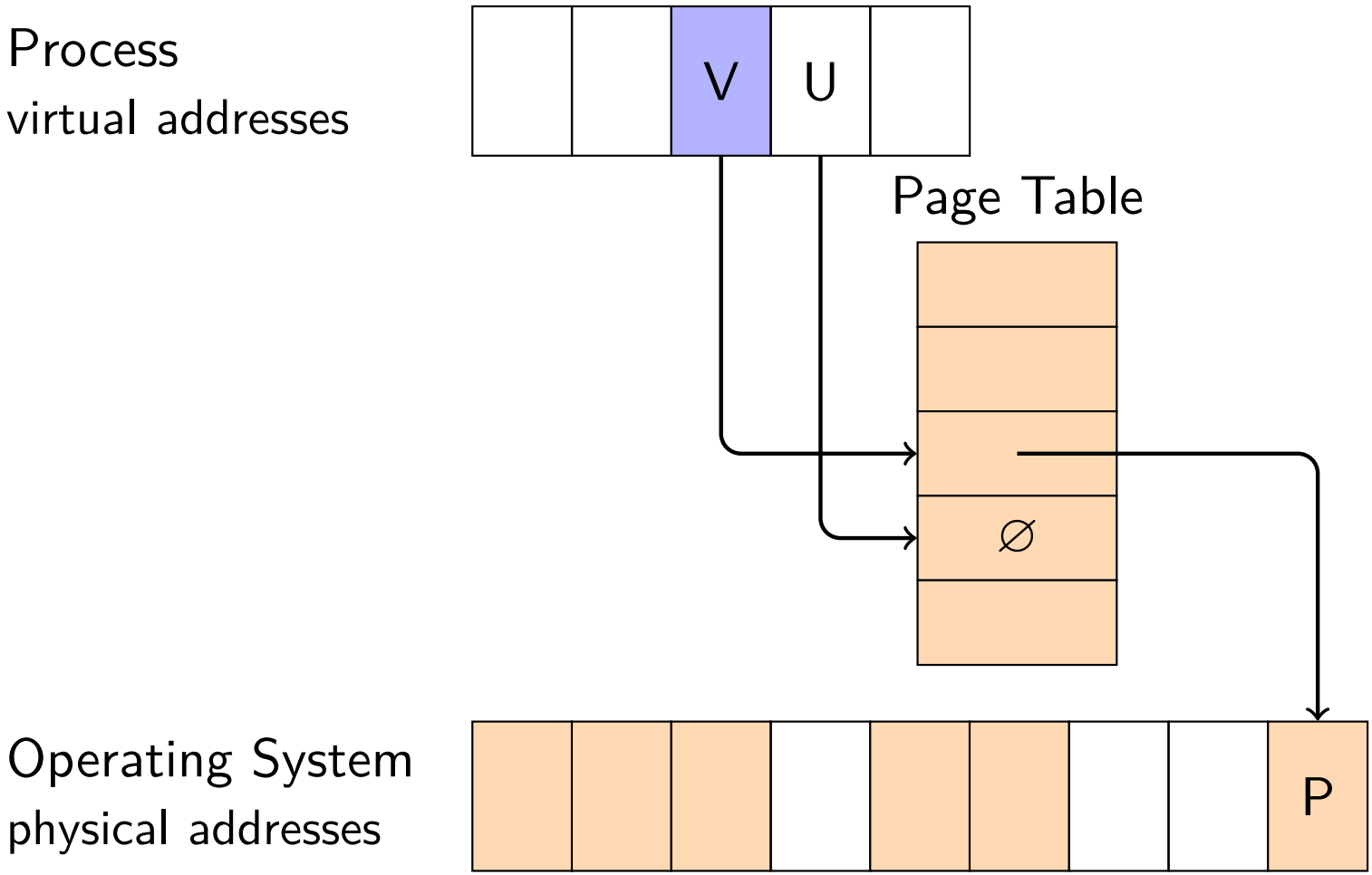
NUMA performance criteria

- Accessing data from local memory
- Avoiding to create contention



How OS mitigate NUMA effects

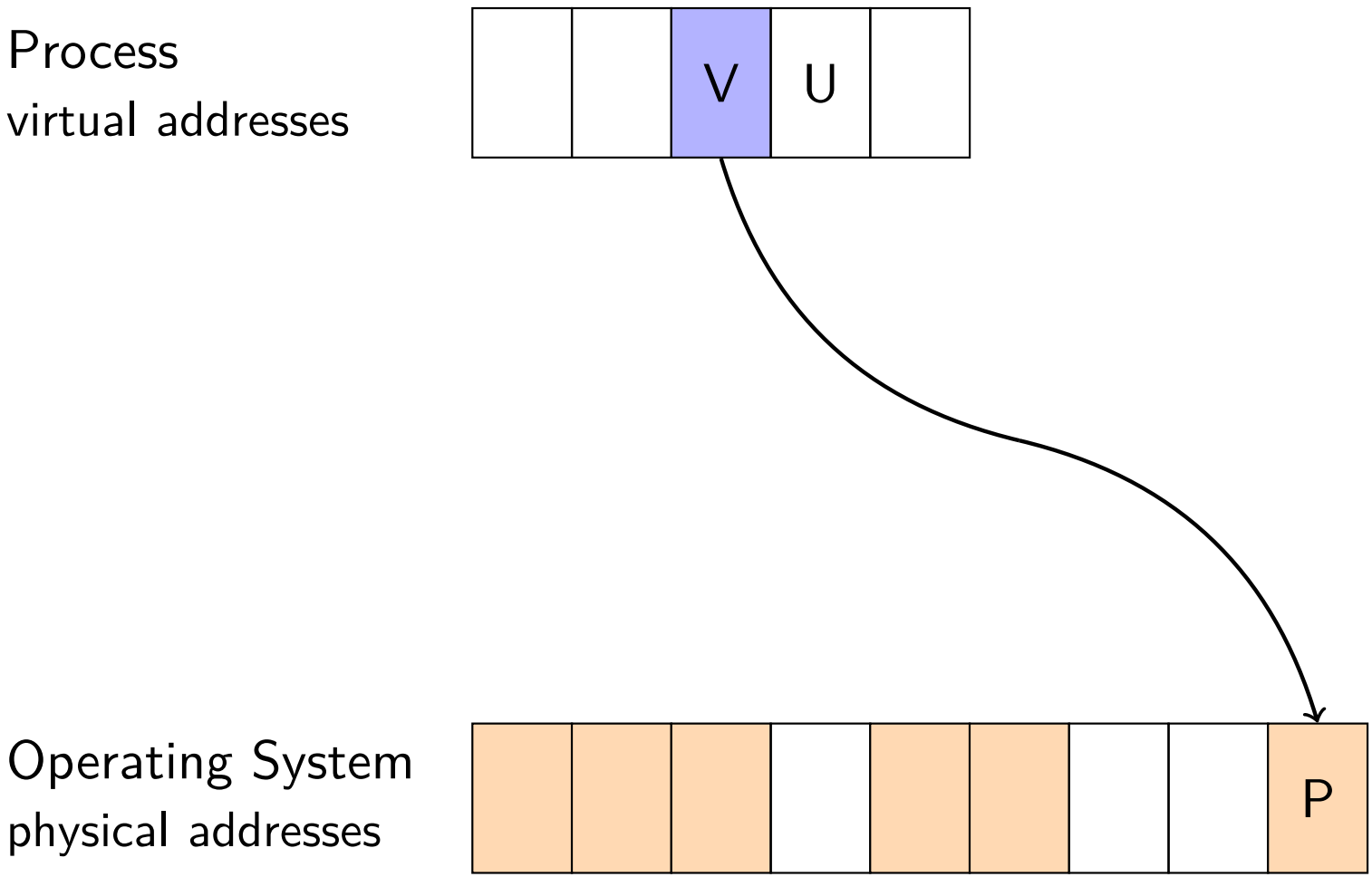
By leveraging the page table of the process



- Illustration with the First Touch policy

How OS mitigate NUMA effects

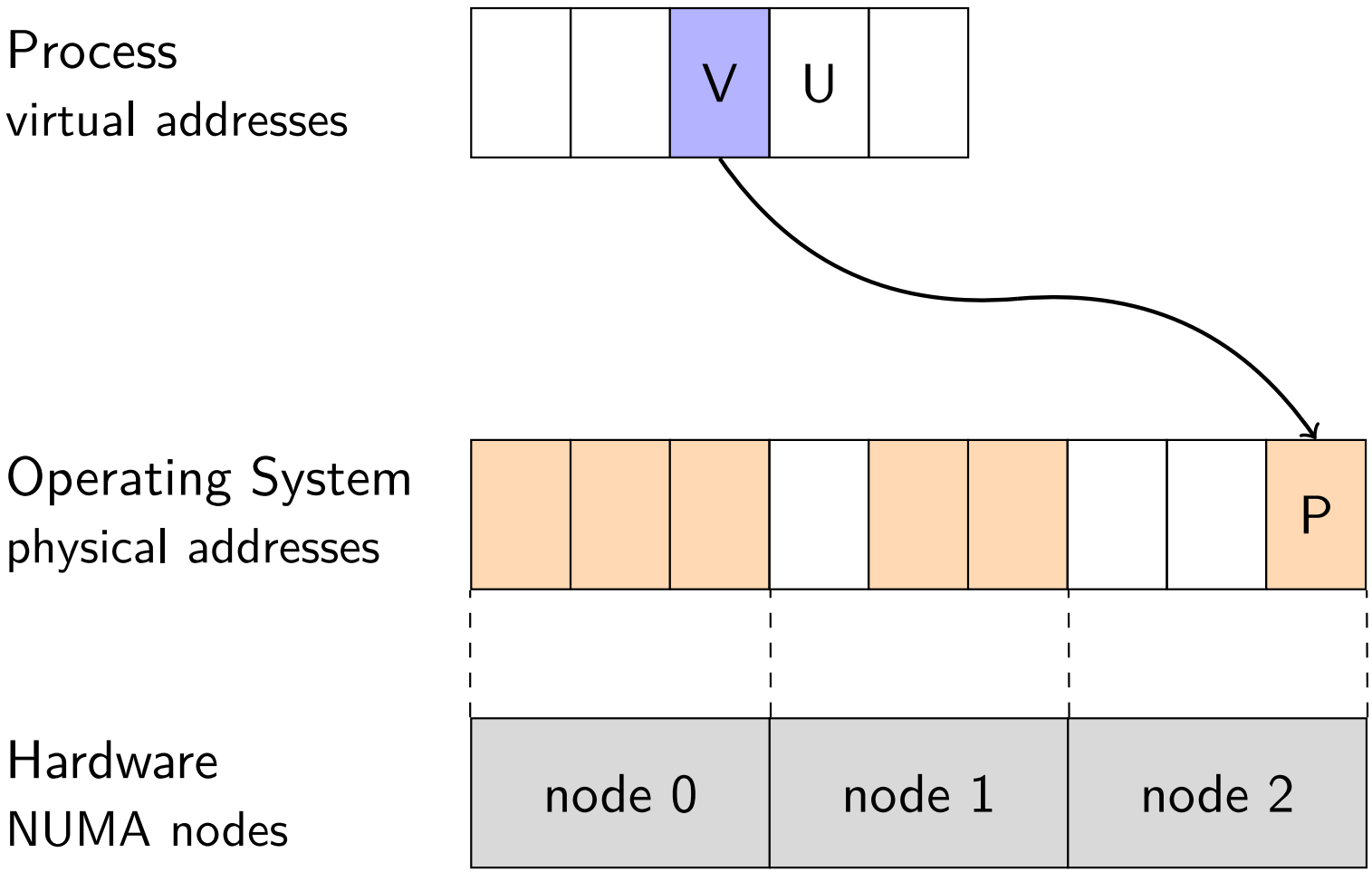
By leveraging the page table of the process



- Illustration with the First Touch policy

How OS mitigate NUMA effects

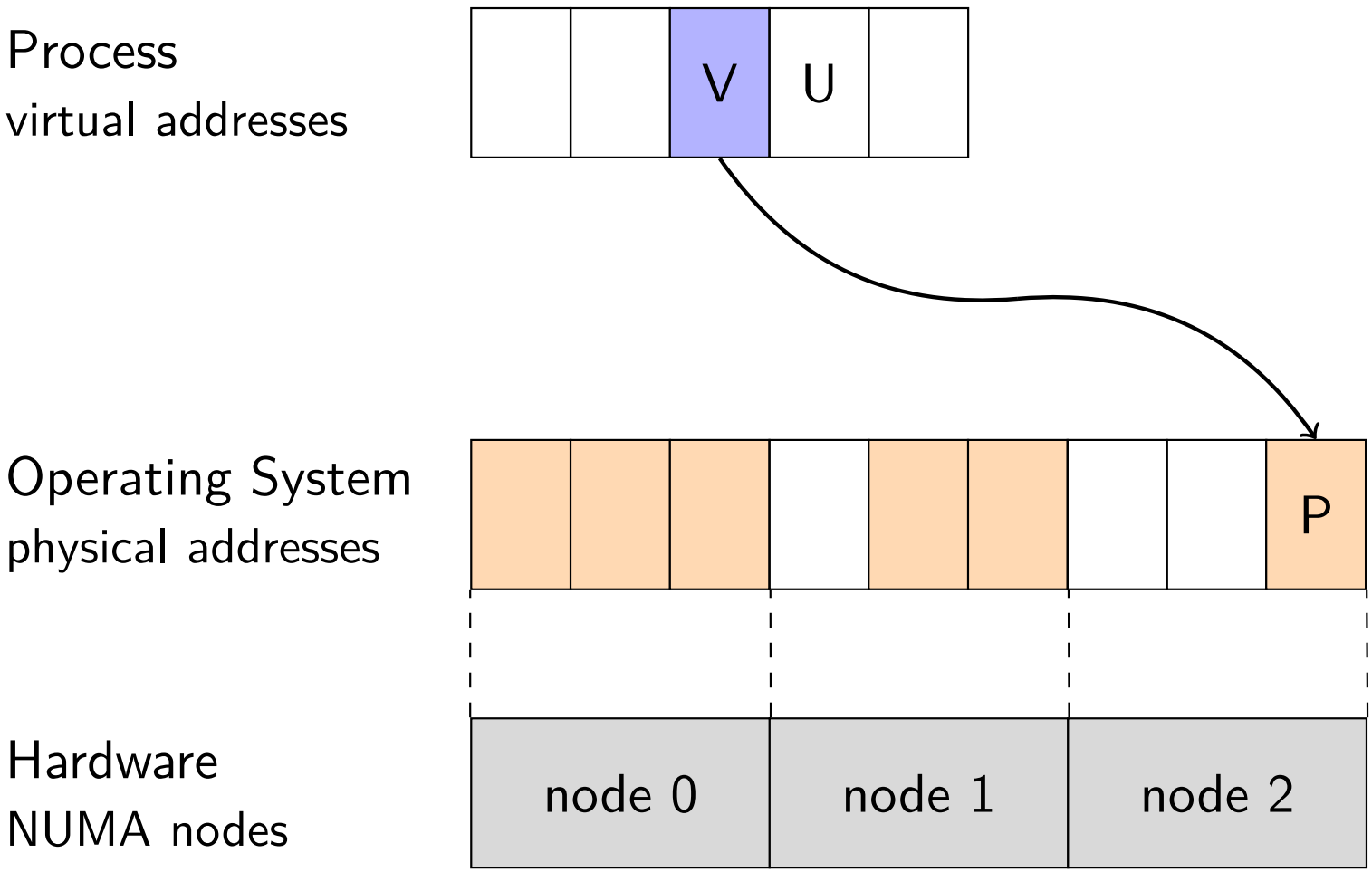
By leveraging the page table of the process



- Illustration with the First Touch policy

How OS mitigate NUMA effects

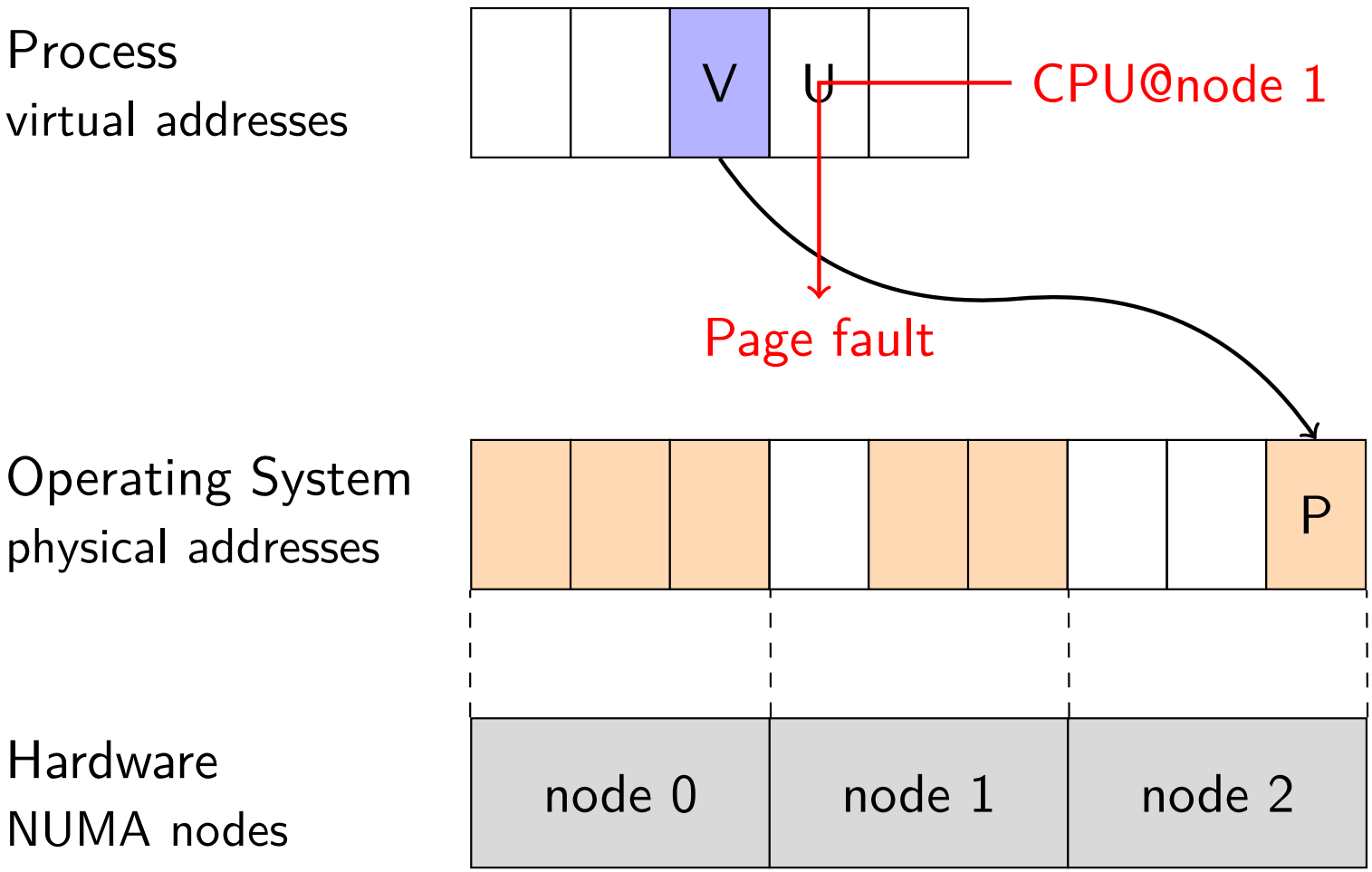
By leveraging the page table of the process



- Illustration with the First Touch policy

How OS mitigate NUMA effects

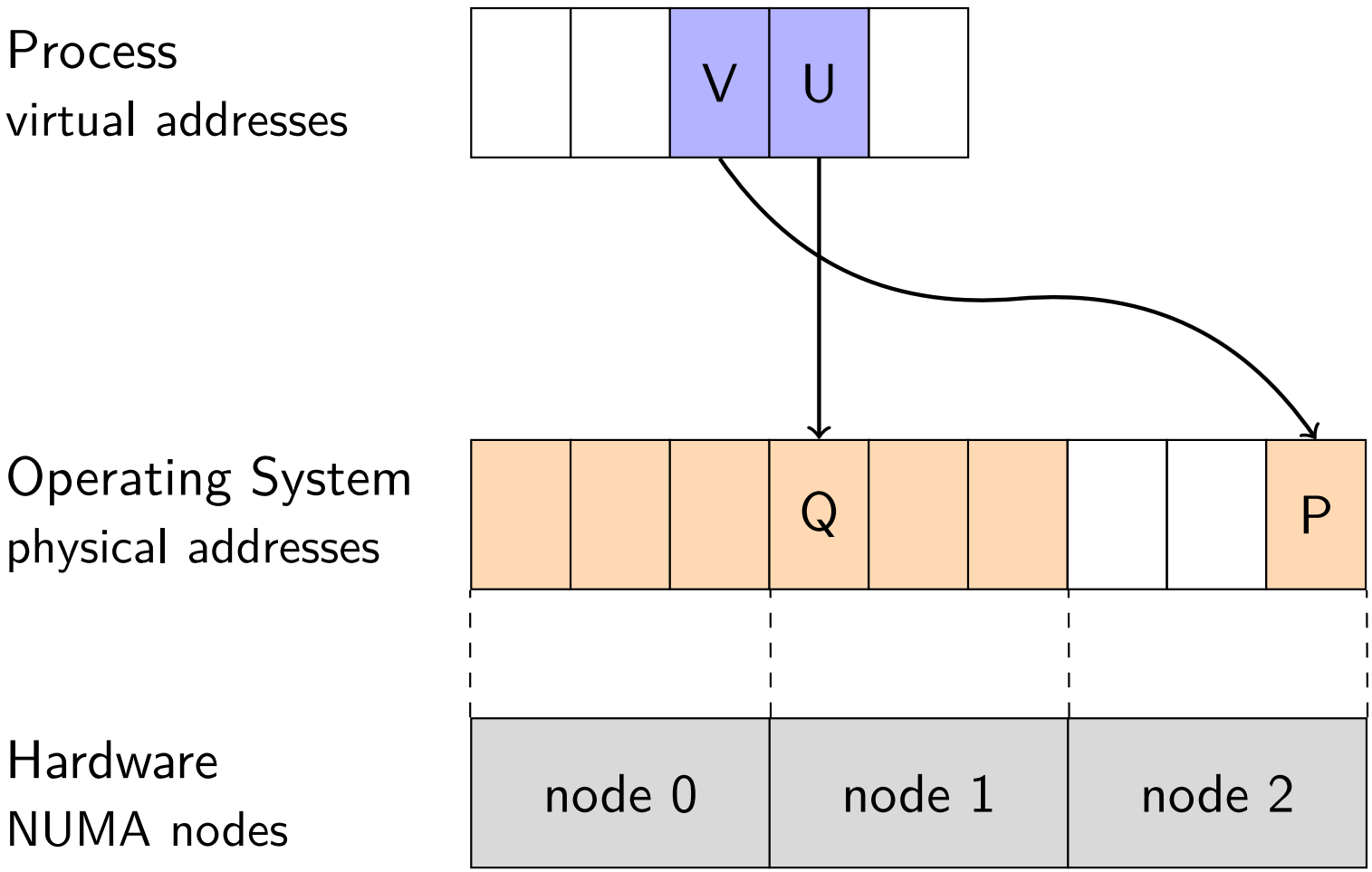
By leveraging the page table of the process



- Illustration with the First Touch policy

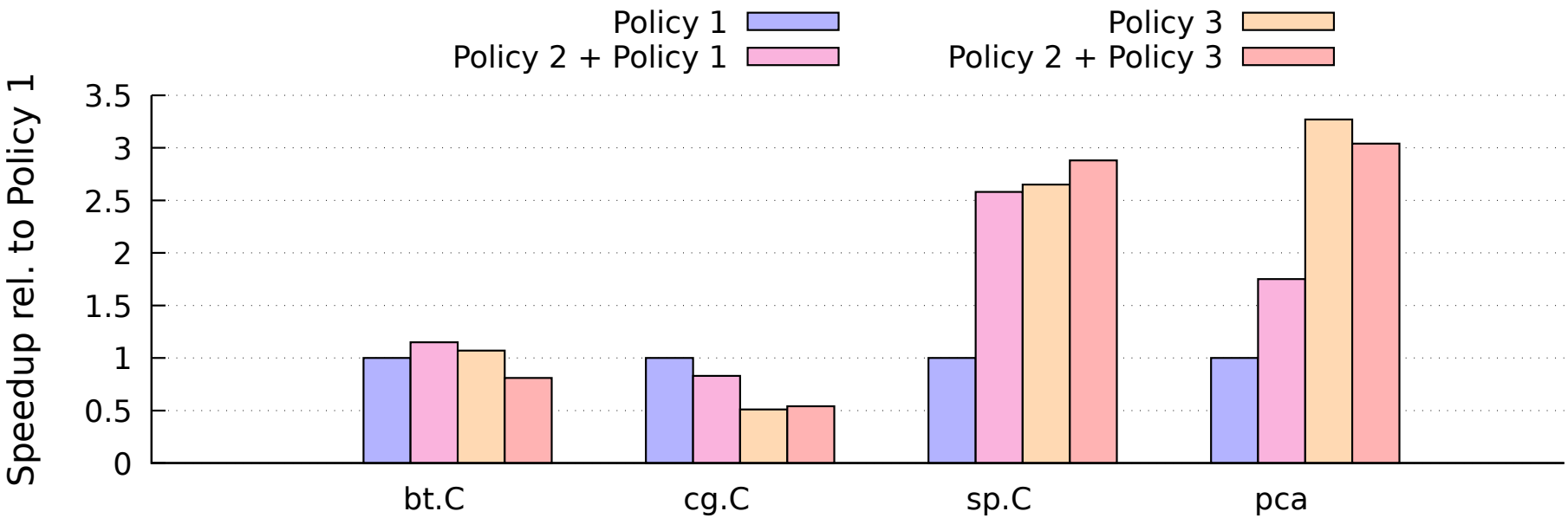
How OS mitigate NUMA effects

By leveraging the page table of the process



- Illustration with the First Touch policy

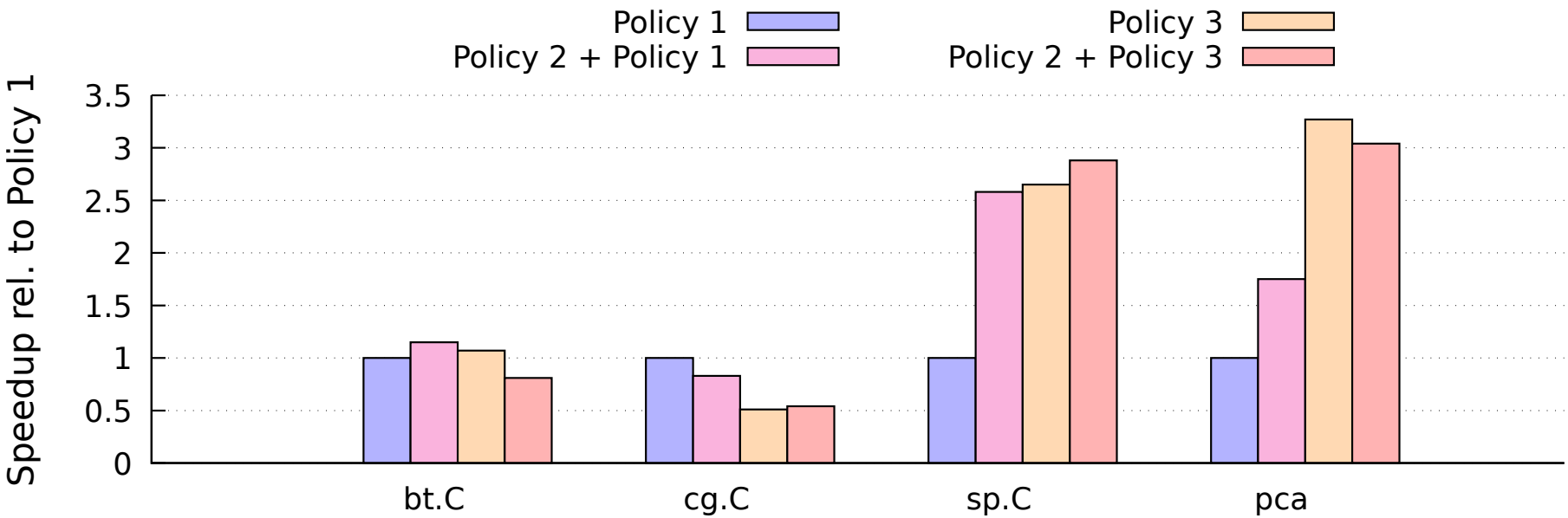
Using NUMA requires smart memory placement policies



- There is not a single best policy for all applications
- Several policies can be the best for different applications
- The policy must be selected depending on the workload

- Operating Systems provide several policies to mitigate NUMA effects
  - But today we use cloud infrastructures

Using NUMA requires smart memory placement policies



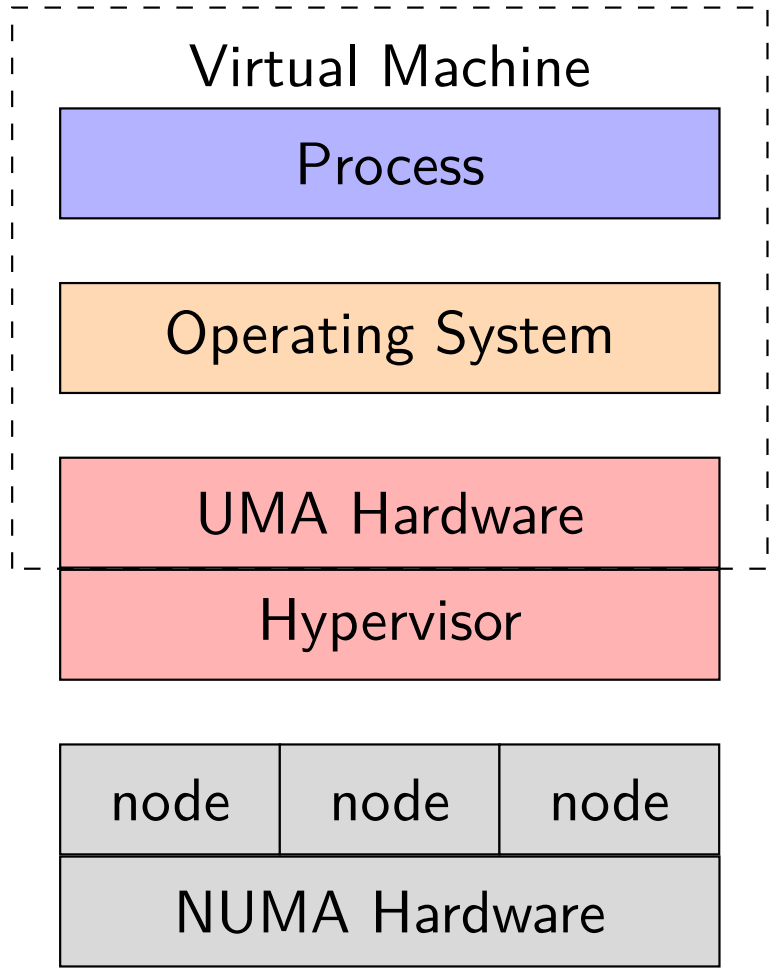
- There is not a single best policy for all applications
- Several policies can be the best for different applications
- The policy must be selected depending on the workload

- Operating Systems provide several policies to mitigate NUMA effects
  - But today we use cloud infrastructures

Cloud infrastructures add a virtualization layer

- Cloud infrastructures rely on virtual machines
- Additional layer : the Hypervisor
  - Manage the virtual machines
- Hides the hardware to the OS

➔ NUMA placement is impossible in the guest Operating System



Contribution

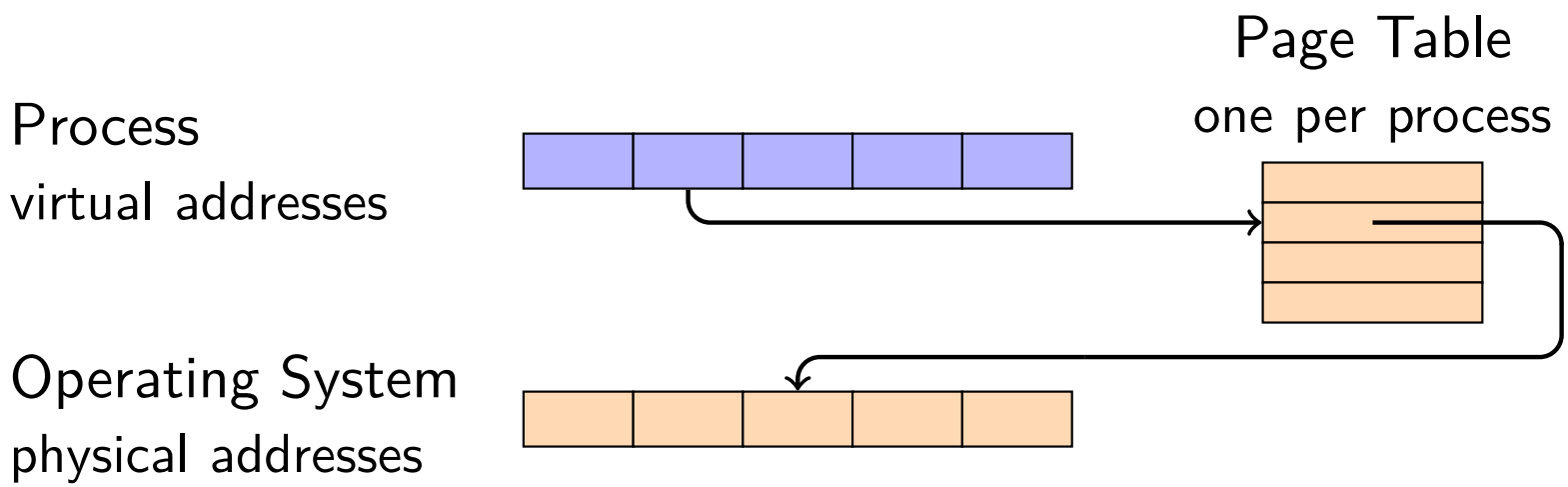
- We move the NUMA management from the guest Operating System to the Hypervisor

- We implement in Xen three efficient policies already used in Linux
  - First Touch
  - Interleaving
  - Carrefour

- Xen places data on the hardware via nested paging
  - We implement policies by using this capability

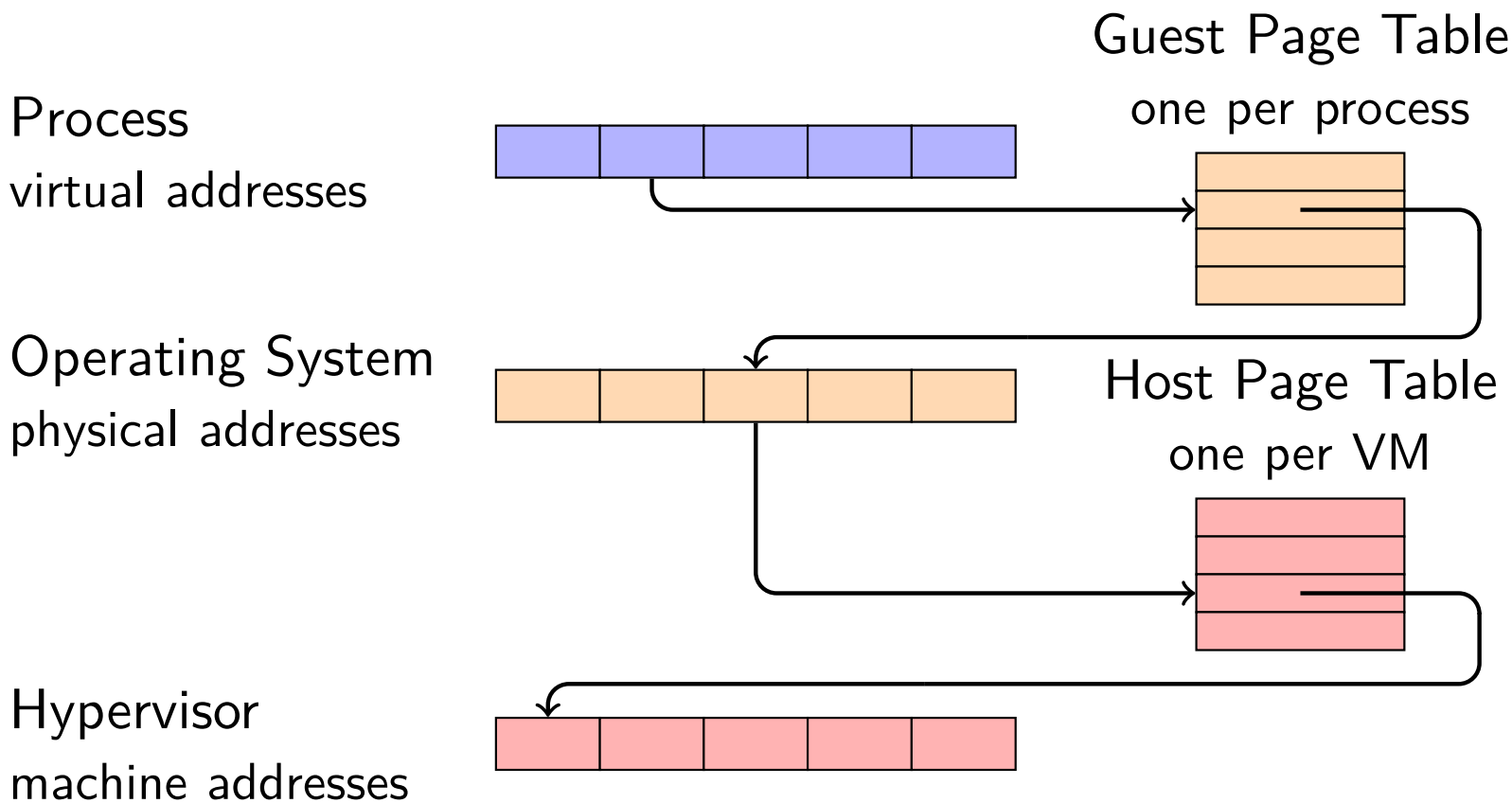
Policies are implemented by using nested paging

- The Operating System places Processes data via the page table



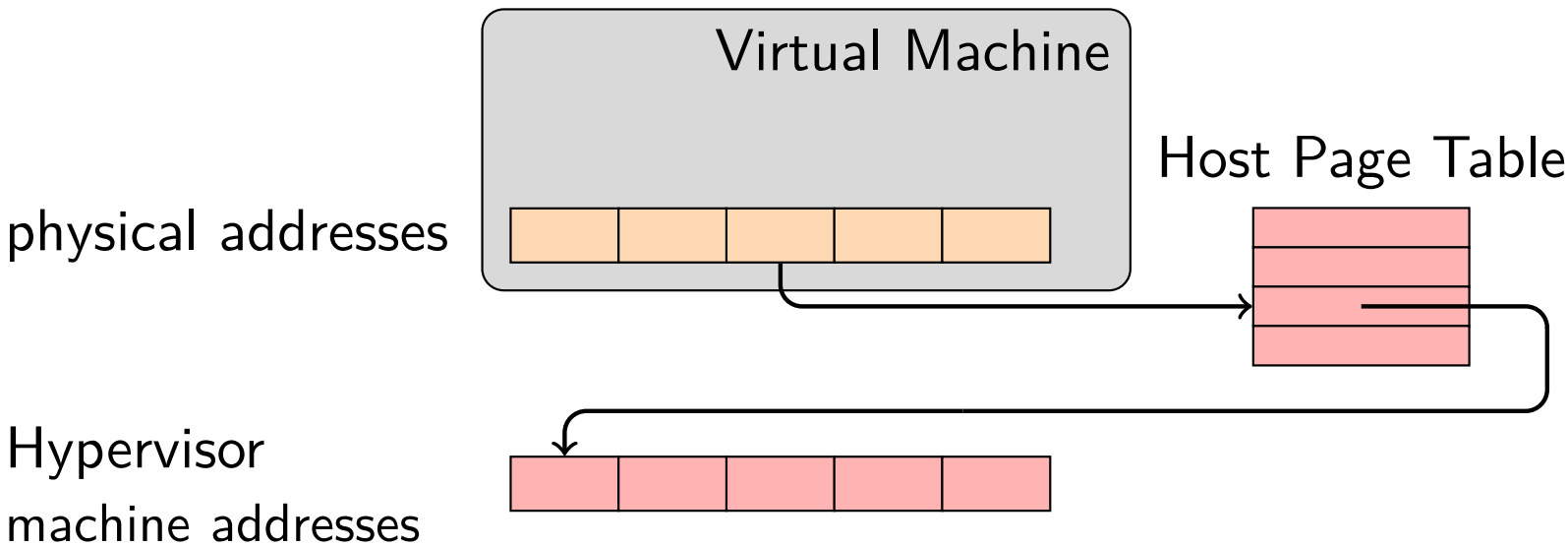
Policies are implemented by using nested paging

- The Operating System places Processes data via the page table
- The Hypervisor overrides this placement with an additional translation
- The two translations are independant



Policy implementation details : First Touch

- Principle : allocate process memory on the first accessor node
- Issue : the hypervisor does not see first accesses
  - Process page faults are hidden to the hypervisor



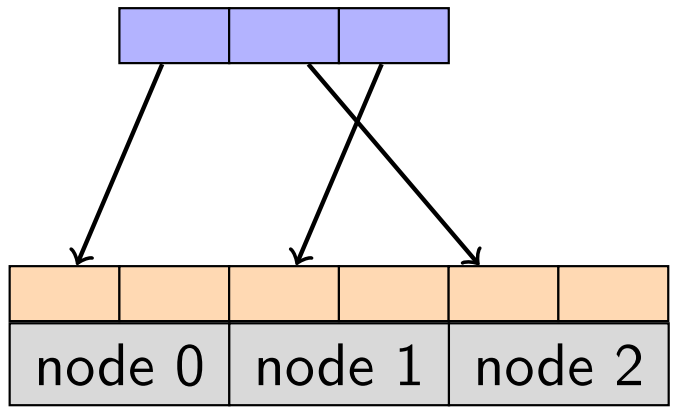
- Approach : paravirtualization

Policy implementation details : Interleaving

- Principle : allocate process memory on a random node
- Issue : the hypervisor does not see process address space
- Approach : interleave the entire physical space of the VM

Process
virtual addresses

Operating System
physical addresses



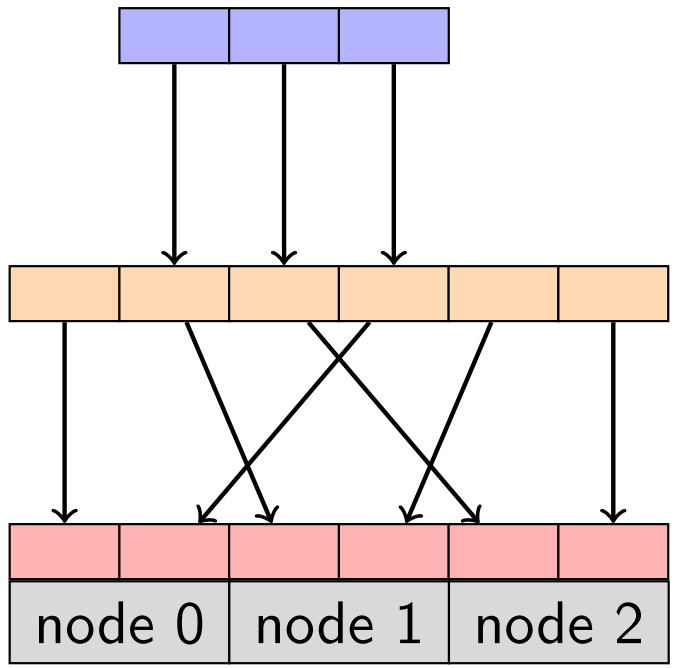
Policy implementation details : Interleaving

- Principle : allocate process memory on a random node
- Issue : the hypervisor does not see process address space
- Approach : interleave the entire physical space of the VM

Process
virtual addresses

Operating System
physical addresses

Hypervisor
machine addresses

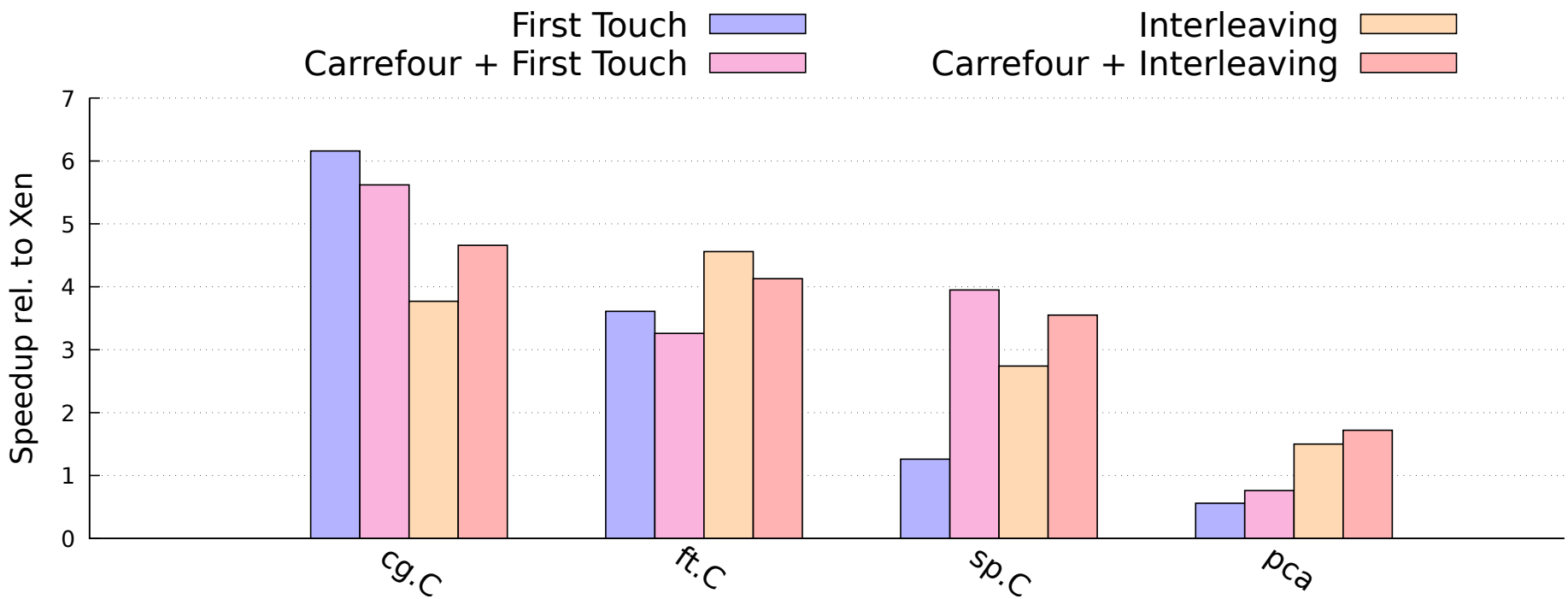


Policy implementation details : Carrefour

- Principle : detect problematic pages and migrate them
  - Use hardware counters to find misplaced pages
- Issue : hardware counters behave differently with nested paging
  - Indicate virtual addresses whereas Carrefour needs physical addresses
  - Need to walk the guest page table concurrently to the OS
- Approach : lockless optimistic translation
  - Carrefour only needs a statistical vision of memory
  - The translation can be best effort

Our implementation improve the performances of NUMA VM

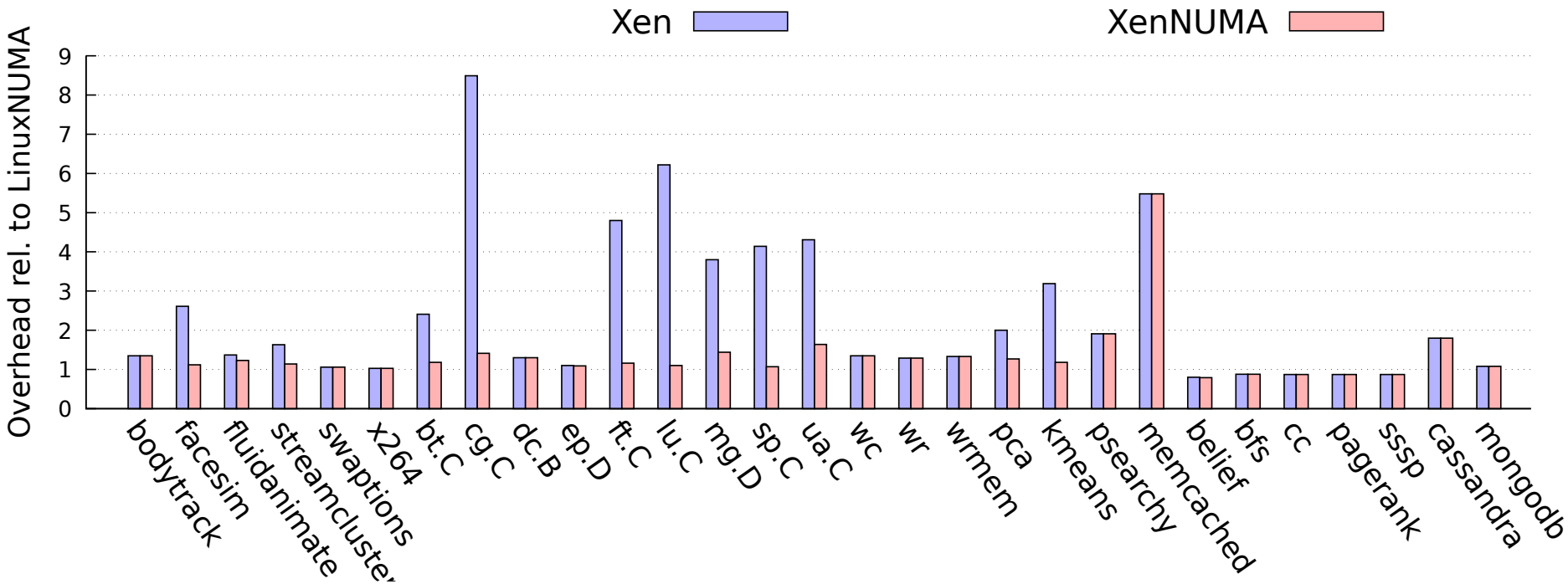
- System : Linux 3.9 / Xen 4.5
- Hardware : 8 nodes / 48 cores / 128 GiB RAM



- Up to 500% speedup compared to default Xen policy
- Each implemented policy can be the best one

Our implementation remove NUMA virtualization overhead

- Software : 29 applications / 5 benchmark suites
- LinuxNUMA : Linux with best NUMA policy
- XenNUMA : Xen with best NUMA policy



- Applications with large overhead : 14 $\rightarrow$ 4
- Average overhead : 139% $\rightarrow$ 35%

Conclusion

- We move the NUMA management from the Operating System to the Hypervisor
- Using common NUMA policies greatly improves performance of applications exectuted in NUMA virtual machines
- Using NUMA policies in hypervisor allows to reach native performance in NUMA virtual machines
- Thank you

Conclusion

- We move the NUMA management from the Operating System to the Hypervisor
- Using common NUMA policies greatly improves performance of applications exectuted in NUMA virtual machines
- Using NUMA policies in hypervisor allows to reach native performance in NUMA virtual machines
- Thank you