Virtual Memory to Support Unified Address Space

**Why support a unified address space?**
- Significantly improves applicability of NMAs
  - Allows NMAs to operate within the same process as CPU
- Enables “pointer-is-a-pointer” semantics
  - Any pointer can be dereferenced on CPU or NMAs
  - Each element has a single name in the address space
  - No explicit copies between CPU and NMAs are required

**Virtual Memory’s Associativity**

- Virtual memory is essentially fully associative:
  - “Any virtual page can be mapped to any physical frame”
- Any mapping is possible → Cannot access memory before translating

**DIPTA: Distributed Inverted Page Table**

- Each page can be placed at a single or few possible locations (set associative)
  - The set is statically determined by the virtual address
  - Location is known without translation (still need to check the page table)
  - Breaks “translate-then-fetch” dependency

**Observation-I:** To fully eliminate translation overhead, translation should finish before data fetch
**Observation-II:** Data and translation always needed together

- Co-locate page tables with data:
  - Fetch, move (and cache) together!

**Results:** Up to 4.9x performance improvement over TLB-based translation