LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation

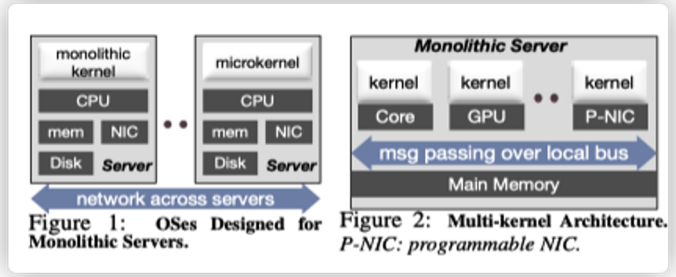
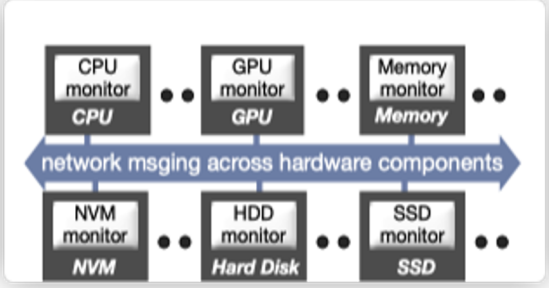


Figure 1: OSes Designed for Monolithic Servers.

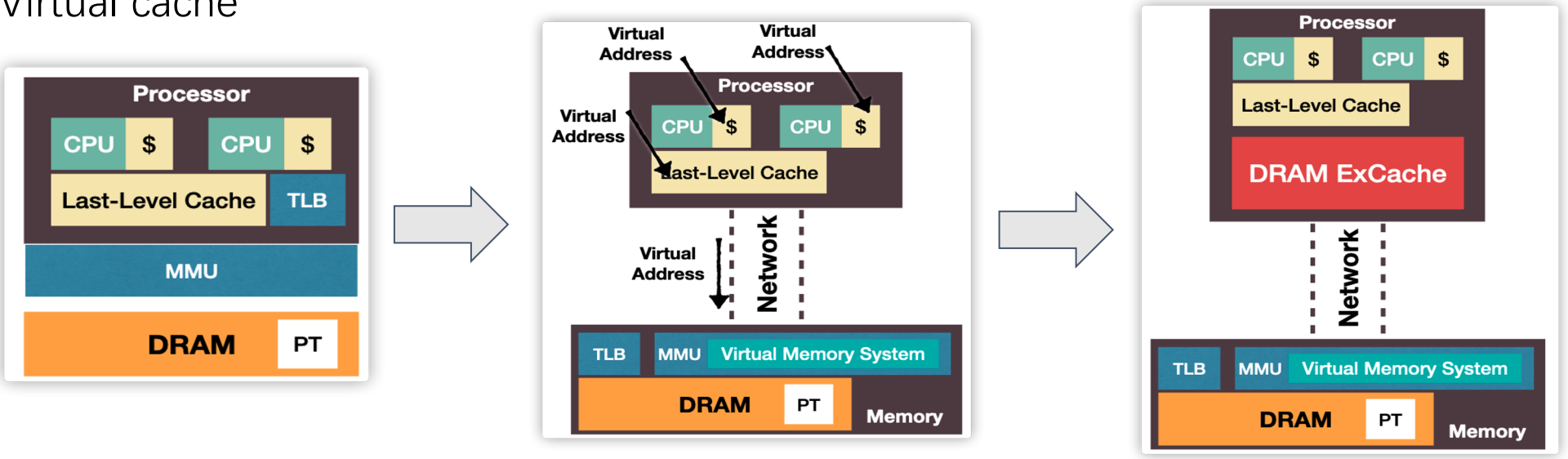
Figure 2: Multi-kernel Architecture. *P-NIC: programmable NIC.*

- Motivation
  - Data center **cannot utilize full** memory and CPU resource
  - It is **difficult** to add, remove, or change **hardware components**
  - Existing kernel designs cannot address the new challenges hardware resource disaggregation brings
  - How to manage and virtualize the distributed, disaggregated hardware components?
- LegoOS design
  - **Clean separation** of OS and hardware functionalities (Basic)
  - Build **monitor** with hardware constraints (not clear?)
  - **RDMA**-based message passing for both kernel and applications (just use RDMA)
  - Two-level distributed resource management (like MPK?)
  - Memory failure tolerance through **replication** (a normal design)



1. Clean separation of OS and hardware functionalities

- Processor components only see virtual memory addresses
  - All levels of cache are virtual cache
- Memory components manage virtual and physical memory
- Add small DRAM/HBM at processor
  - Use it as Extended Cache, or ExCache
    - Software and hardware co-managed
    - Inclusive
    - Virtual cache



4. Two-level distributed resource management

- GMM assigns vRegions to mem compone
  - On virtual mem alloc syscalls (e.g., mmap)
  - Make decisions based on global loads
- Owner of a vRegion
  - Fine-grained virtual memory allocation
  - On-demand physical memory allocation
  - Handle memory accesses

Looks like MPK?