Disaggregating Memory with Software-Managed Virtual Cache

Yizhou Shan, Yiying Zhang

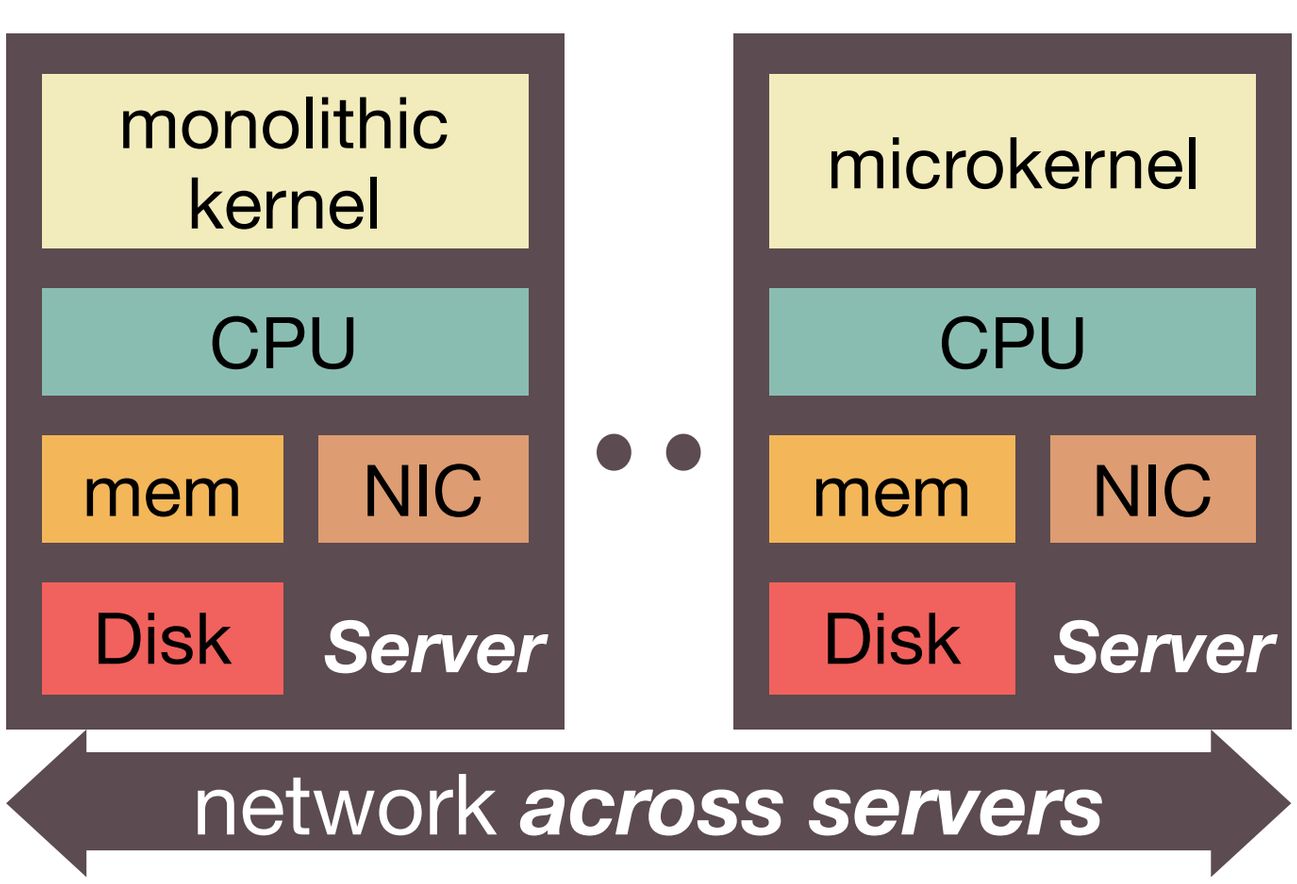
WukLab

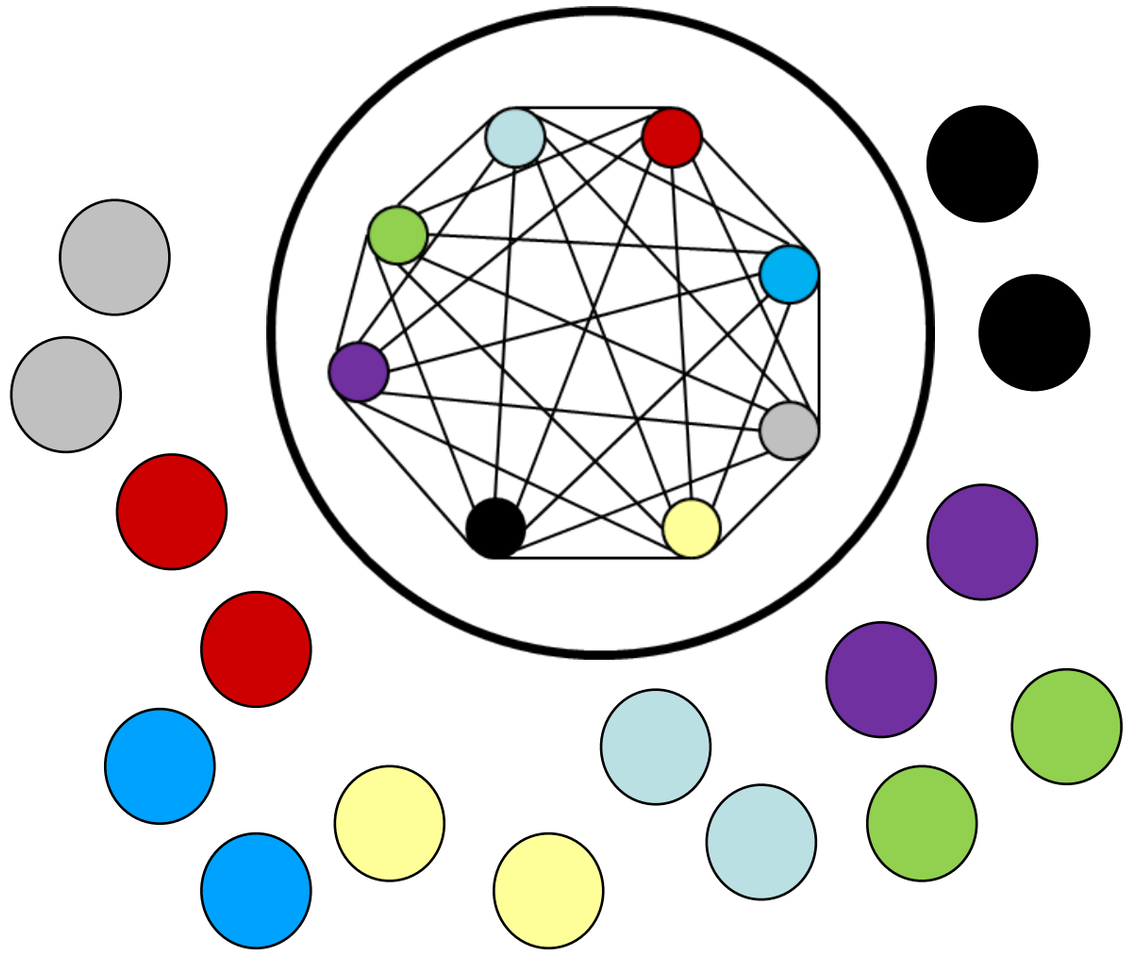
PURDUE UNIVERSITY

Resource Disaggregation:

Breaking monolithic servers into network-attached, independent hardware components

Traditional OSes





- Manages single node and all hardware resources in it
- Bad for hardware heterogeneity and hotplug
- Does not handle component failure

When hardware is disaggregated, the OS should be also!

Lego: the *First* Disaggregated OS

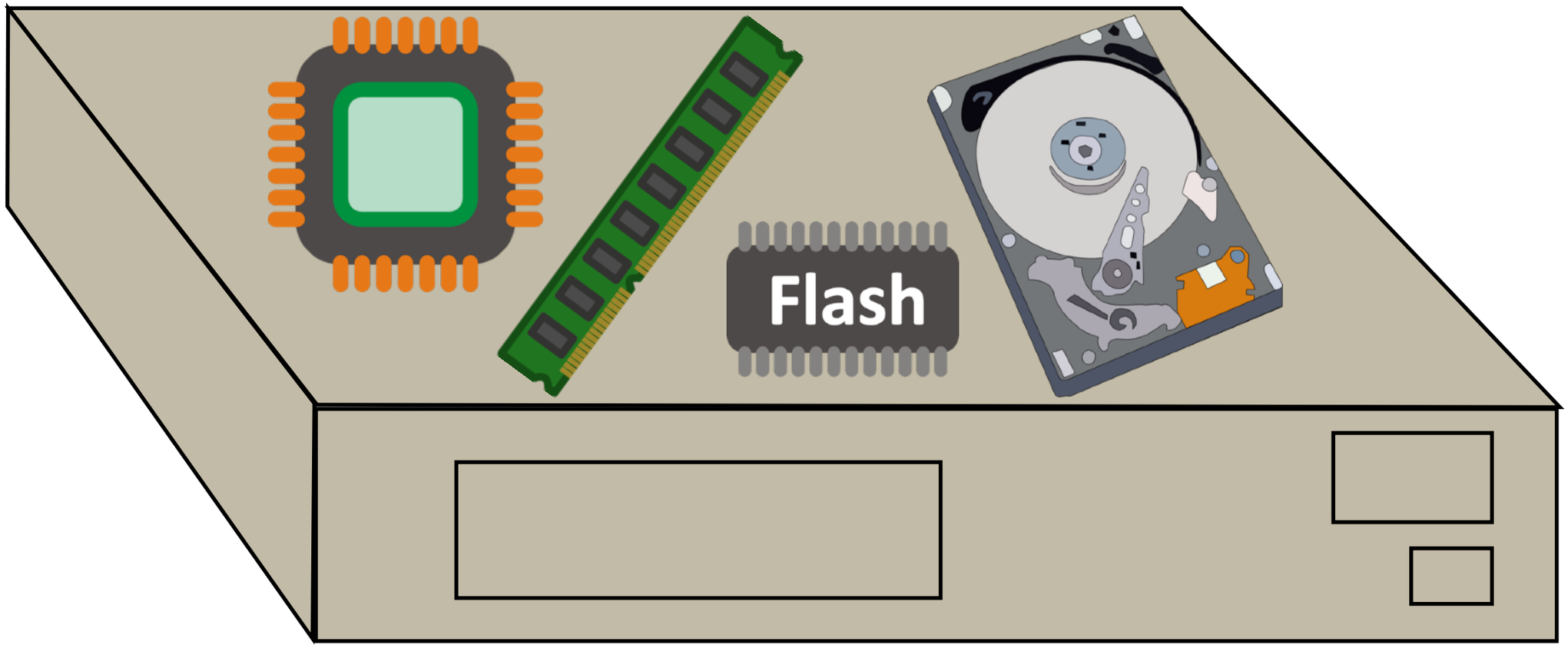
OS

Process Mgmt

Virtual Memory System

File & Storage System

Network



Network

Process Mgmt

Network

Virtual Memory System

Network

File & Storage System

Network

File & Storage System

Network

Flash

Key Challenge: Cost of Crossing Network

	Bandwidth	Latency
Mem Bus	50-100 GB/s	~50ns
PCIe 3.0 (x16)	16 GB/s	~700ns
InfiniBand (EDR)	12.5 GB/s	500ns
InfiniBand (HDR)	25 GB/s	<500ns
GenZ	32-400 GB/s	<100ns

- Network **hardware** is much faster than before
- Current network still slower than local memory bus

Observations in Memory Access and Hardware Trends

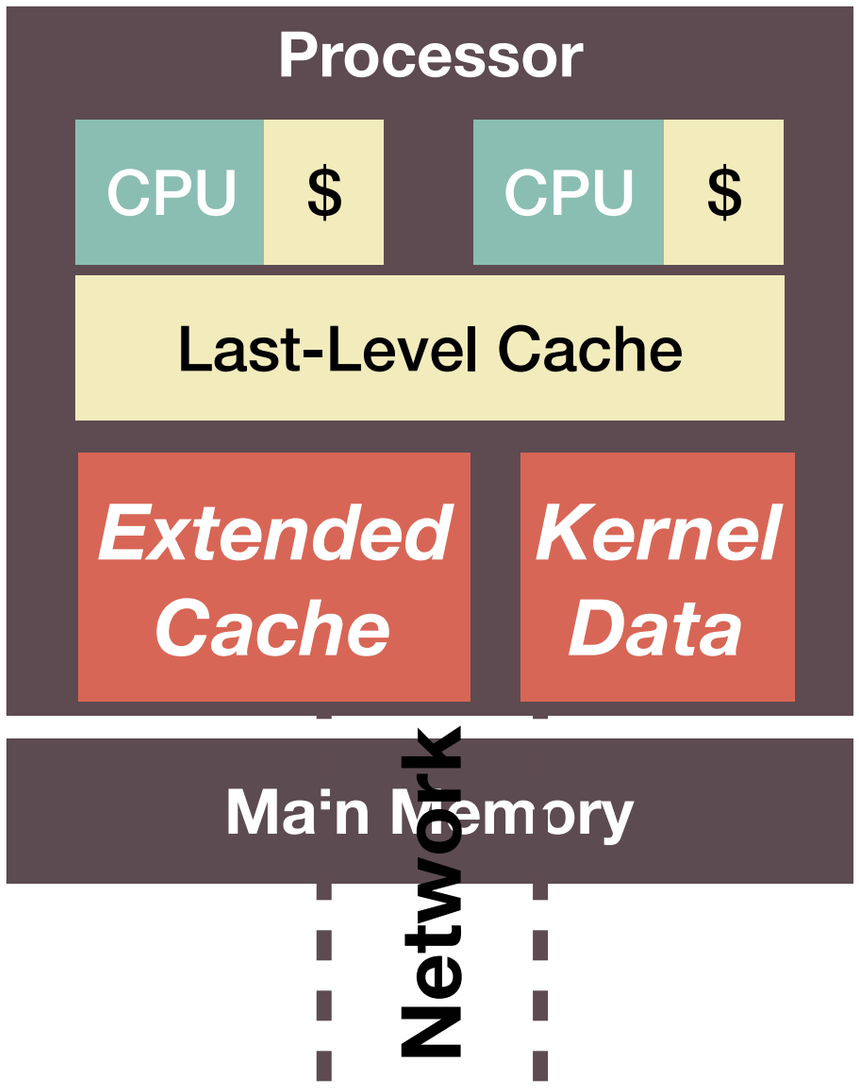
- Total memory footprint can be large,
- but most accesses go to a small portion
  - 90% => 7MB / 9.6GB (pagerank)
  - 95% => 300MB / 9.6GB

- Faster, smaller memory close to CPU
  - HBM, 3D-stacked

- More computation power at memory
  - PIM/PNM

Our Solution: Separate Memory Perf and Capacity

- **Bigger memory behind network**
- **Extended cache at processor**
  - HBM or regular DRAM
  - Can be software-managed
- Separate physical mem for kernel

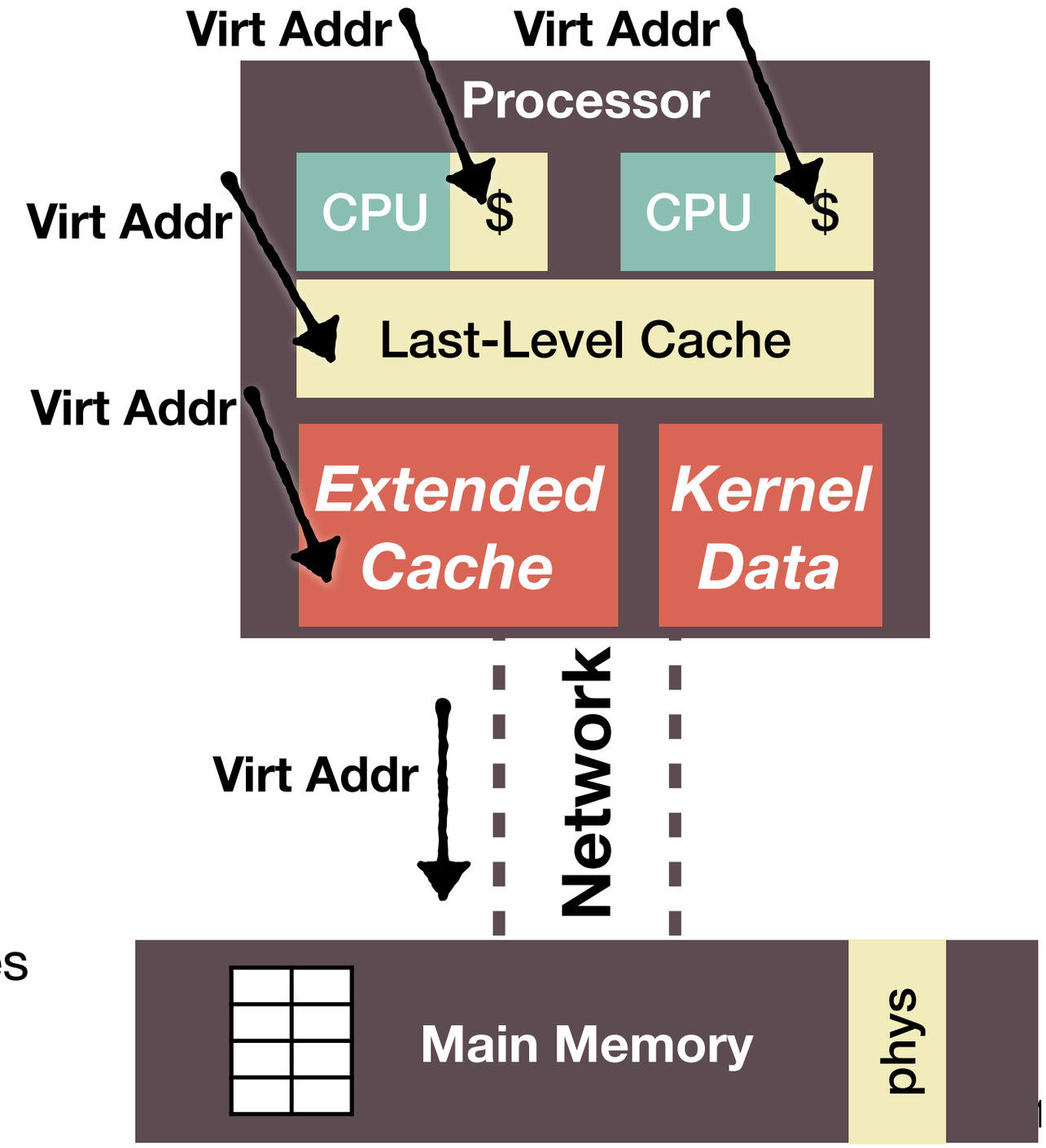


Clean Separation of Processor and Memory Functionalities

- Important for heterogeneity, flexibility, and failure independence

- Memory components manage
  - Virtual and physical memory spaces
  - Virtual to physical memory mapping

- Processors
  - Only see virtual memory addresses
  - Software-managed virtual cache



Other Challenges

- Handling component failure
- Manage distributed, heterogeneous resources
- Fitting micro-OS services in hardware controller
- Implementing Lego on current servers

Implementation and Emulation

- Lego built from scratch, >200K LOC and growing
  - Runs all Linux ABIs and unmodified binaries
  - Manages disaggregated processor, memory, storage
  - Global resource manager
- Hardware emulation
  - Use regular servers to emulate hardware devices
  - DRAM organized as extended cache, managed by page fault handler

Conclusion

- Resource disaggregation calls for new system
- ***Lego***: new OS designed and built for datacenter resource disaggregation
- Separating memory performance and capacity, and processor and memory functionalities
- Many challenges and many potentials

Thank you
Questions?

WukLab

wuklab.io

PURDUE UNIVERSITY