Lego: A Distributed, Decomposed OS for Resource Disaggregation

PURDUE UNIVERSITY

Yizhou Shan, Yilun Chen, Yutong Huang, Sumukh Hallymysore, Yiying Zhang

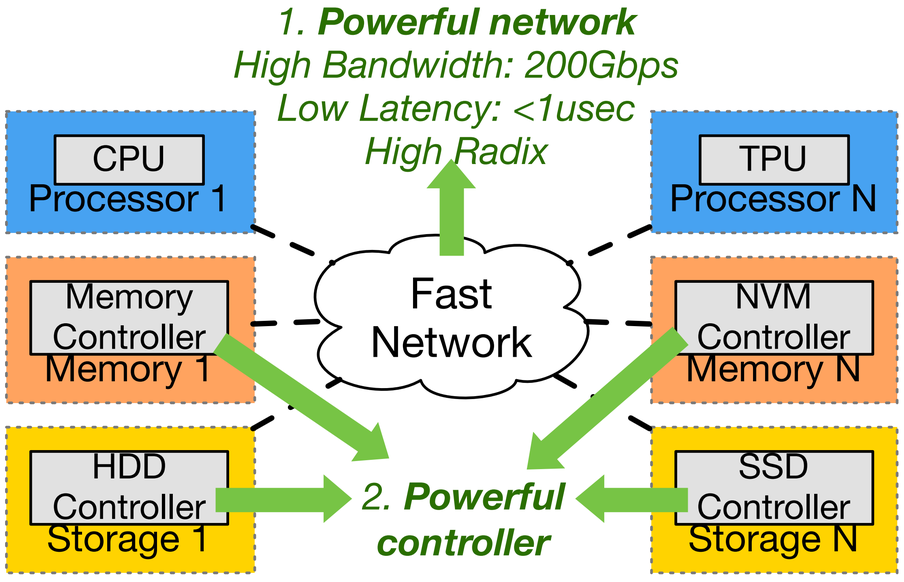
WukLab

WukLab.io

When hardware is disaggregated, the OS should be also!

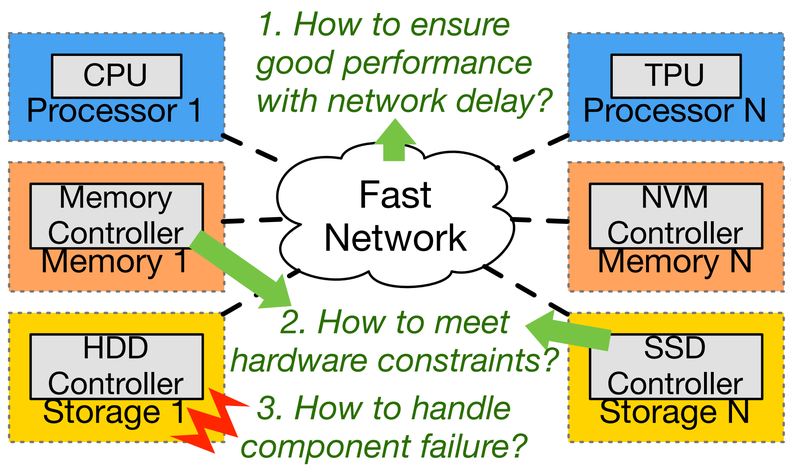
Resource Disaggregation

Breaking *monolithic* servers into network-attached, independent hardware components



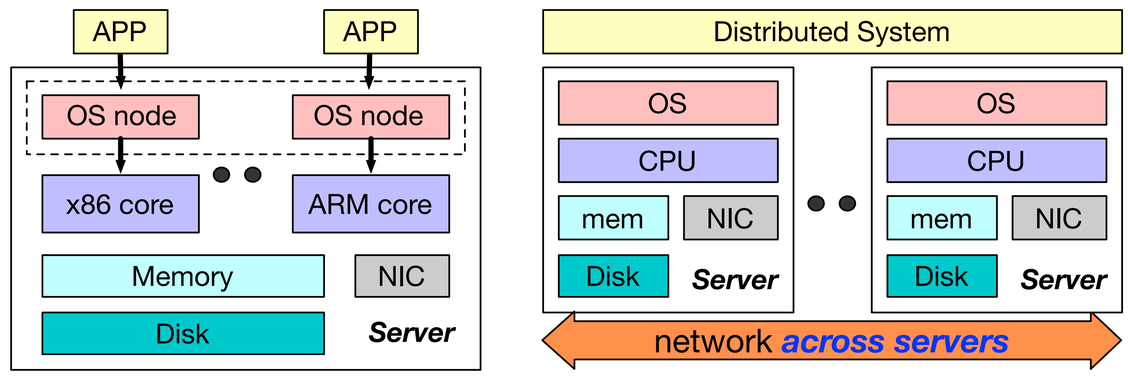
- Efficient resource utilization
- Finer-grained failure domain
- Easy to add/remove/change hardware

Unique Challenges



No existing OS solves the unique challenges of resource disaggregation!

Problems with Existing OSes



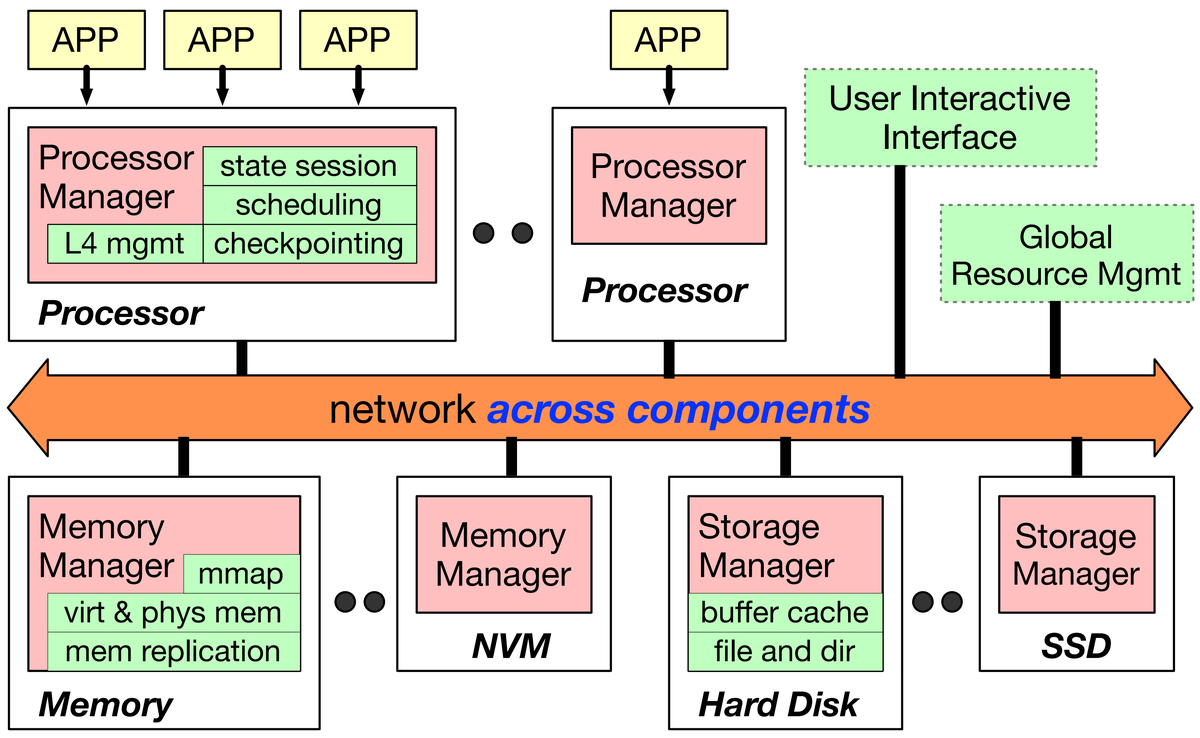
Multikernel: manages cores separately, not other components

Monolithic/micro- kernel: built for monolithic server

Distributed OS: manages distributed monolithic servers, but not distributed hardware components

Lego: a Disaggregated OS

How to build an OS for resource disaggregation?



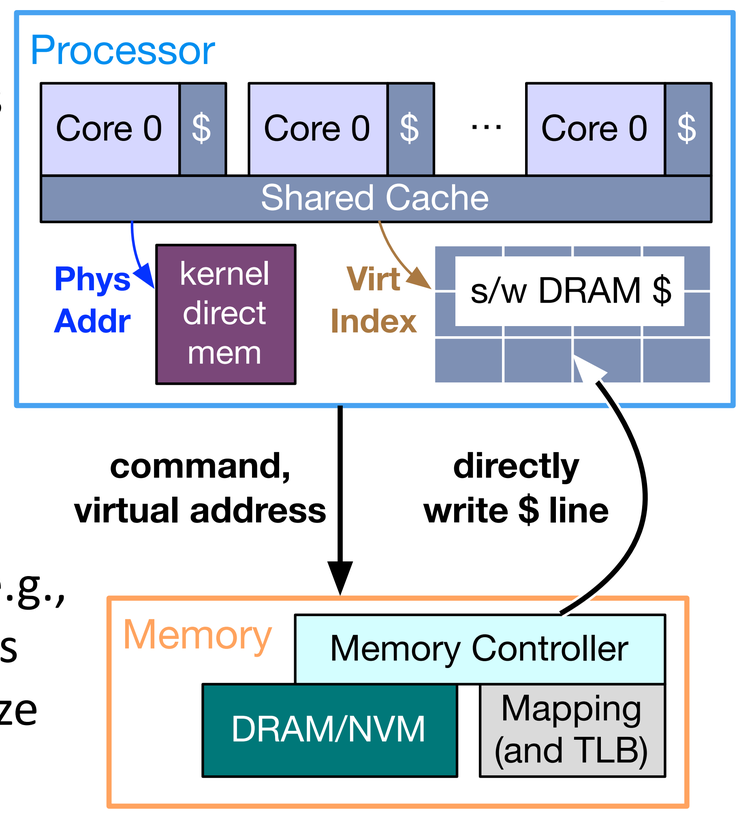
- Split kernel into stateless managers
- Handle failures transparently
- No memory sharing across processors
- Built from scratch; ongoing, open-source project
- Supports x86-64 and unmodified Linux binaries

Hardware Design

How to cleanly separate hardware components while ensuring good performance?

Processor

- Only knows virtual addresses, all caches are virtually indexed and tagged
- Small local DRAM as software-managed last-level cache
- Small physically-addressed memory for kernel usages



Memory/Storage

- Device controllers (e.g., ***ASIC***) run OS services
- Manage and virtualize DRAM/NVM/HDD

Failure Handling

Handle Processor Failure

- Per-process coordinated checkpointing
- Stateless managers → efficient checkpointing
- Minimal dependencies across processes

Handle Memory Failure

- Replicate memory at checkpointing time
- *Selective* dirty memory replication → ***< 2X space***
- Use replicated memory for better parallelism
- *Lazily* compact replicas and checkpoint to storage