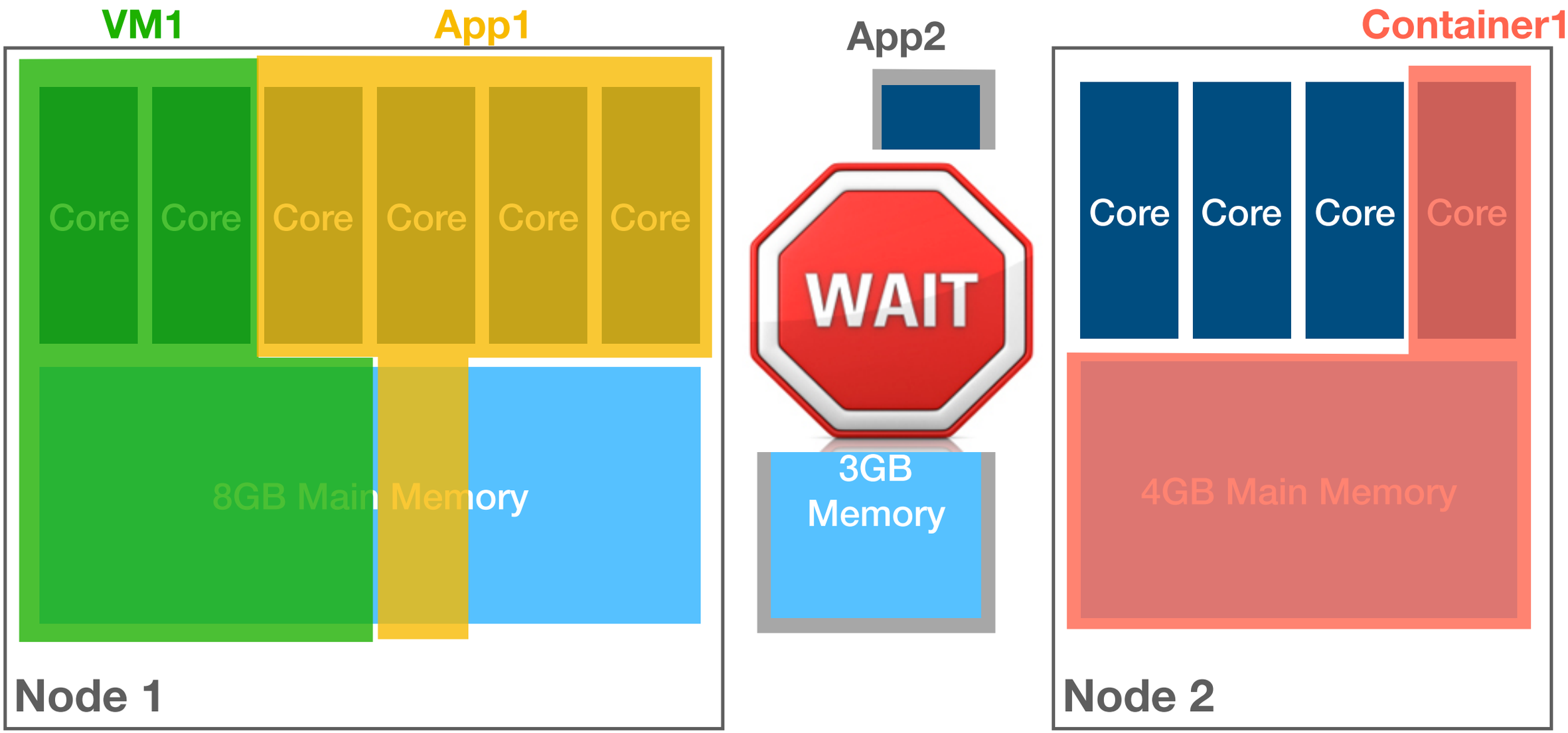
Split Container: Running Containers beyond Physical Machine Boundaries

Yilun Chen, Yiying Zhang

WukLab

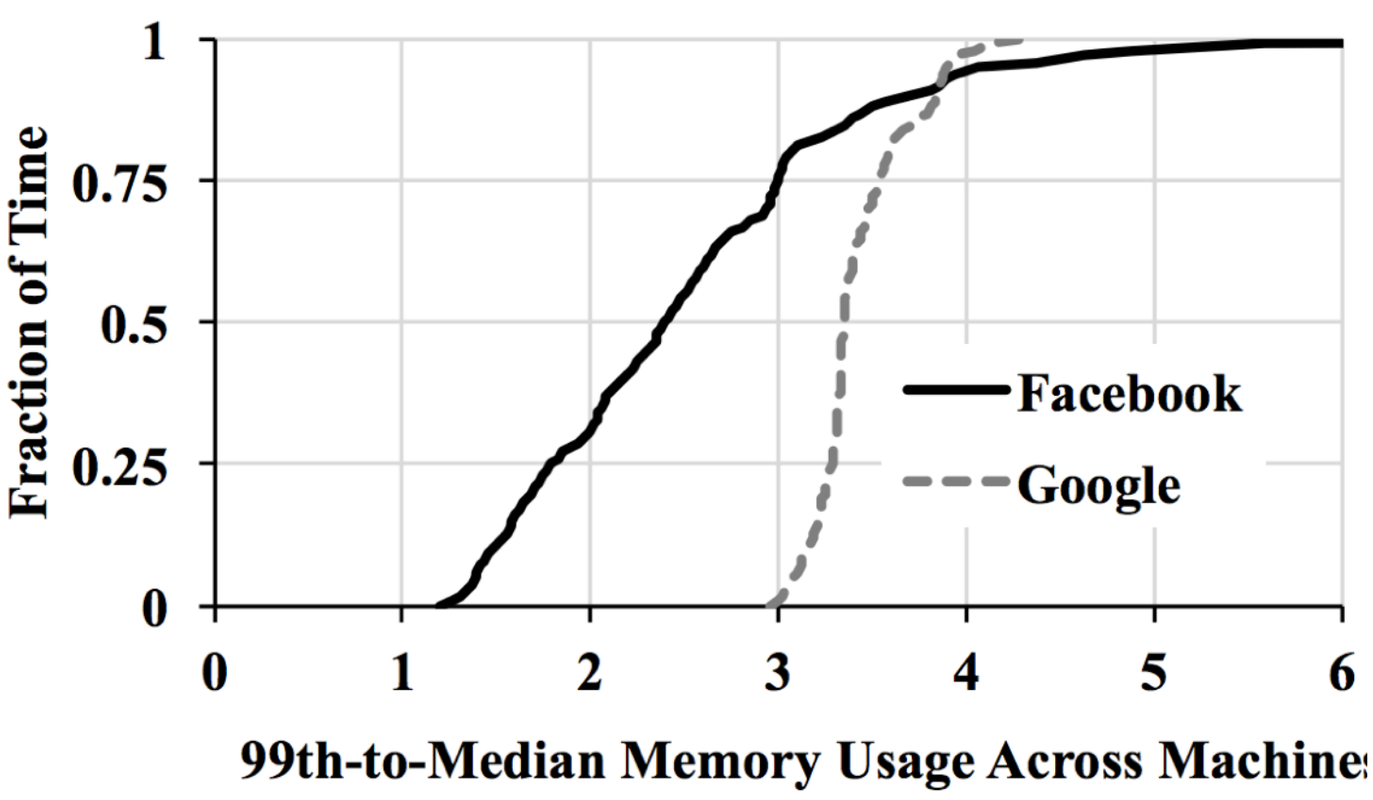
PURDUE UNIVERSITY

Resource Allocation in Datacenters

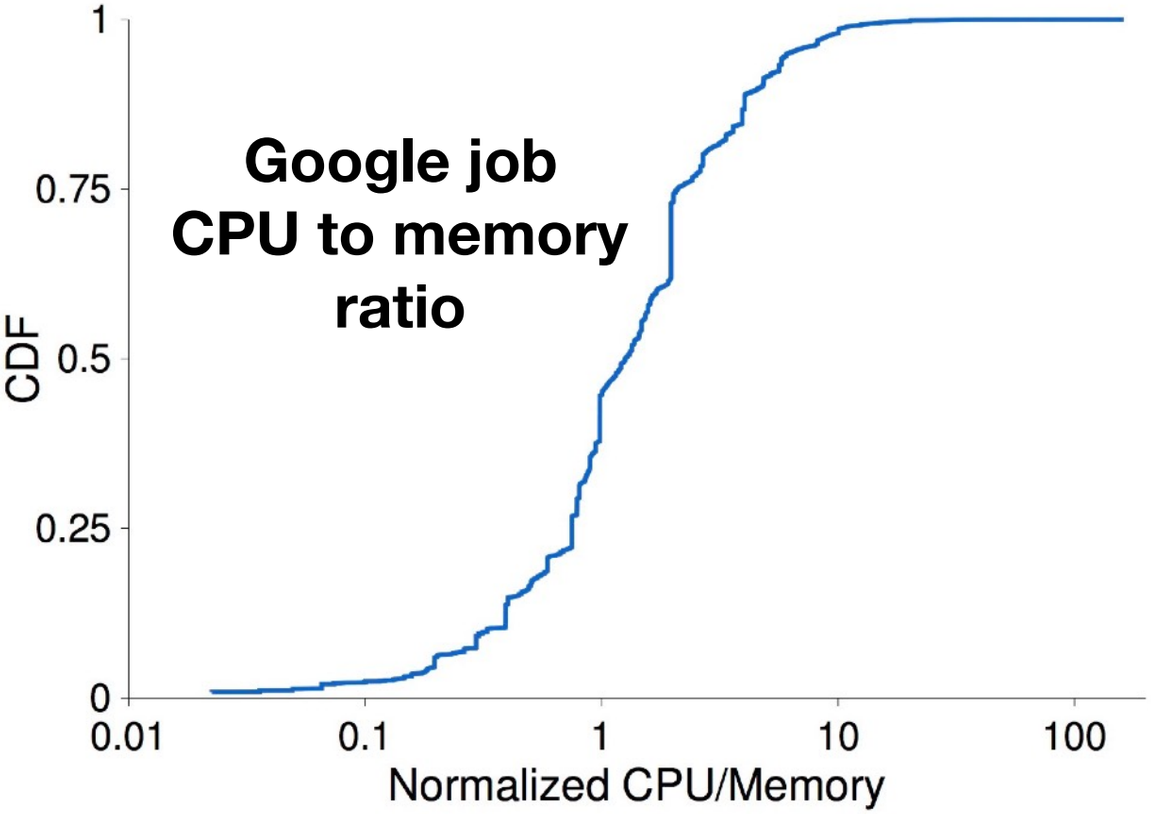


Physical Machine Boundary!

CPU/Memory Usages across Machines and across Jobs



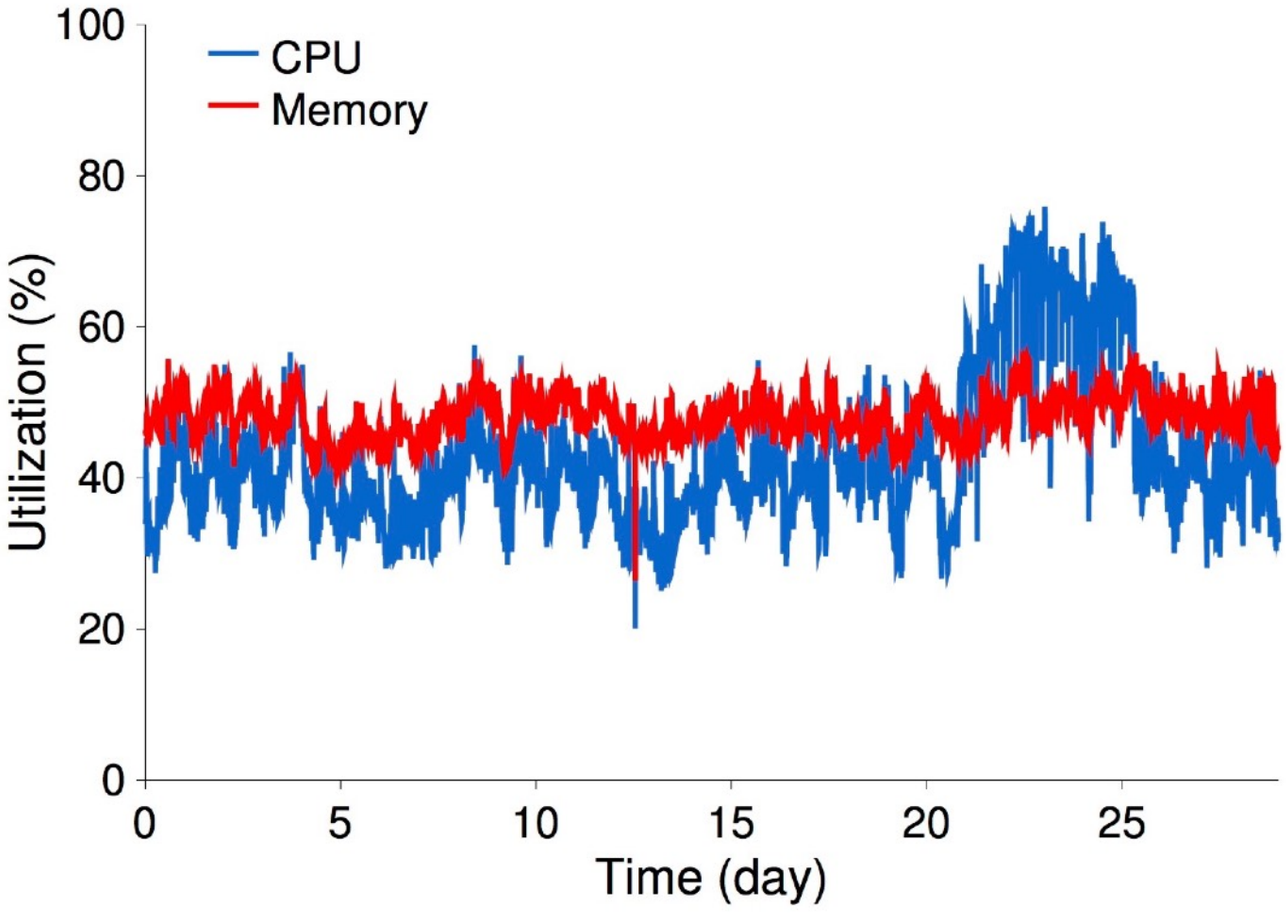
Source: Gu et al. “Efficient Memory Disaggregation with Infiniswap” NSDI’17

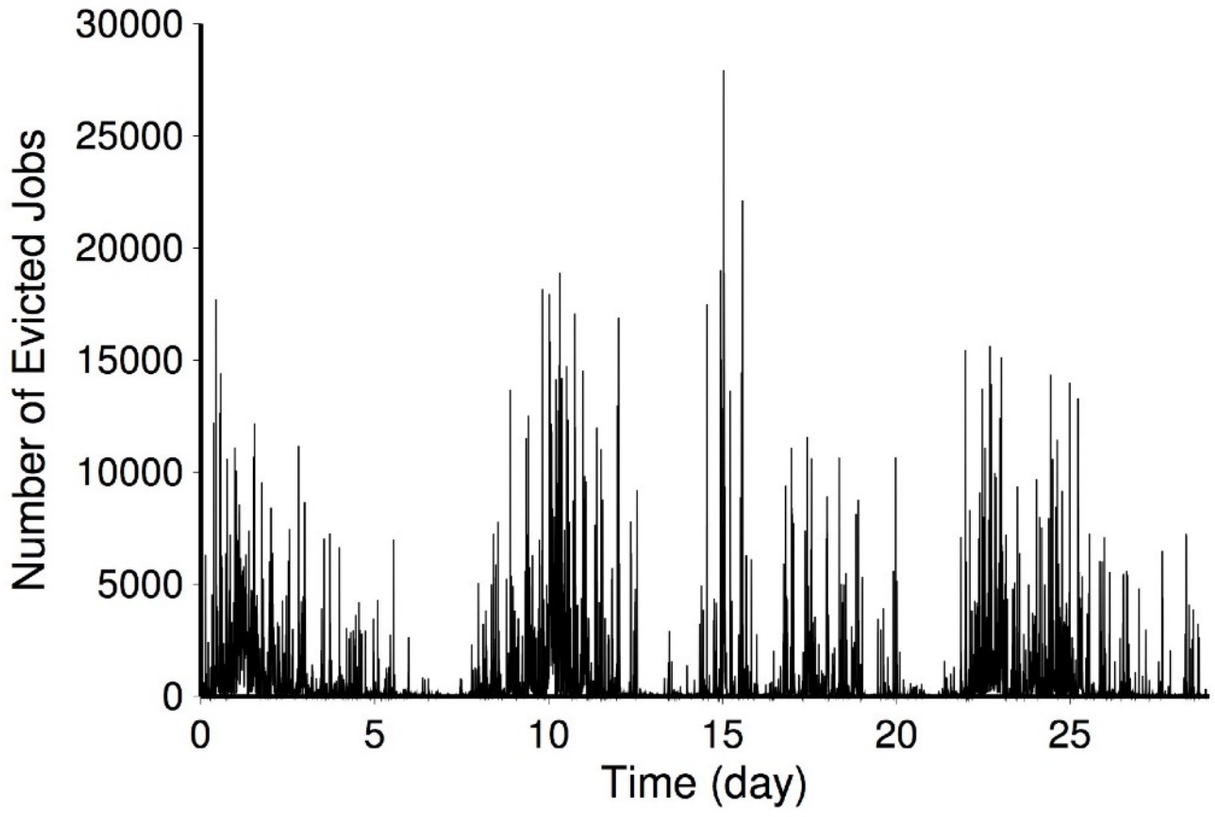


Google Cluster Trace “https://github.com/google/cluster-data”

Modern Datacenter Applications Have Heterogeneous CPU/Memory Requirements

Resource Utilization in Production Clusters





* Google Production Cluster Trace Data. “https://github.com/google/cluster-data”

Unused Resource + Waiting/Killed Jobs Because of Physical-Node Constraints

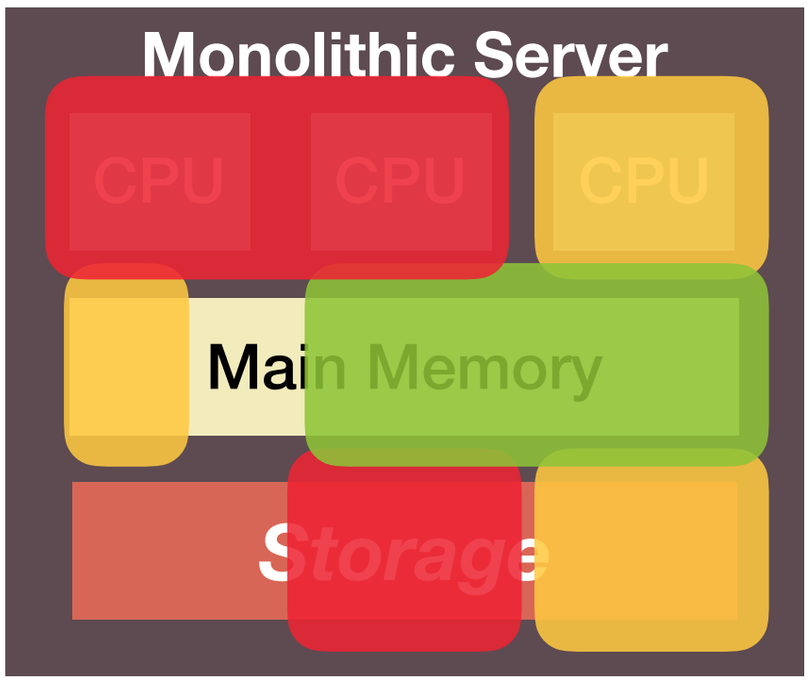
Physical Resource Disaggregation

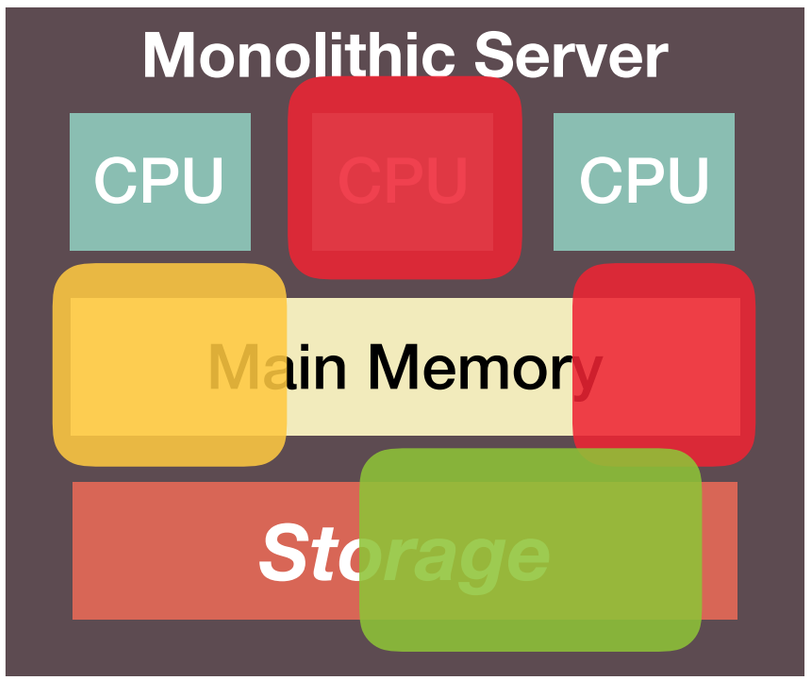
- Great support of heterogeneity
- Very flexible in resource management
- But needs hardware, network, and OS changes

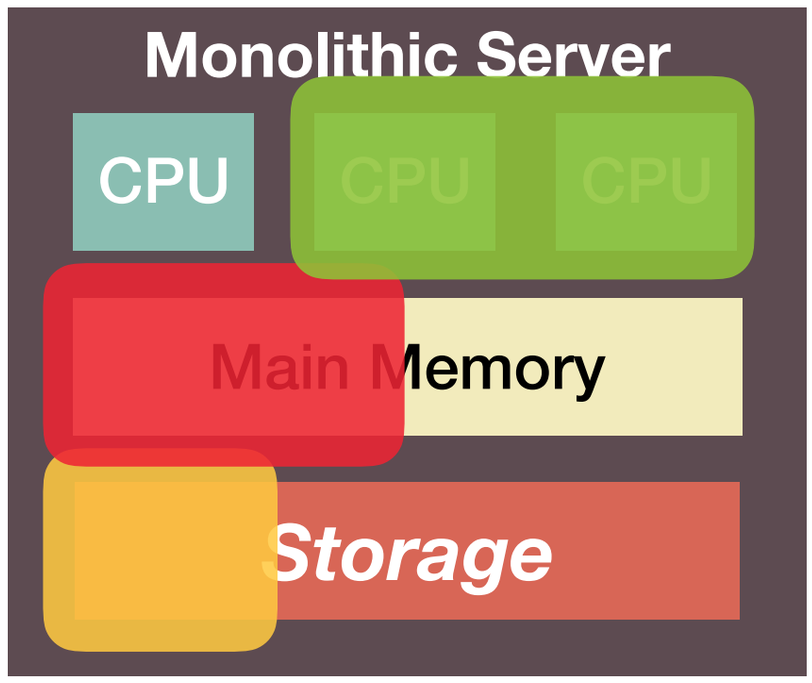
Is there any less disruptive way to achieve better resource utilization and elasticity?

Virtually Disaggregated Datacenter

- Use resources on remote (distributed) machines







Using Remote/Distributed Resources

- Was a popular idea in 90s
  - Remote memory/paging/swap
  - Network block device
  - Distributed shared memory (DSM)

- No production-scale adoption
  - Cost of network communication
  - Coherence traffic

REVISIT YOUR
OLD IDEAS
WITH NEW EYES.

Remote/Distributed Memory in Modern Times

- New application trends
  - Large parallelism
  - New computation and memory requirements
  - New programming models

- Network is 10x-100x faster
  - InfiniBand: 200Gbps, <500ns
  - GenZ: 32-400GB/s, <100ns

Recent New Attempts

- Distributed Shared Memory
  - Grappa
  - Hotpot (Distributed Shared Persistent Memory)
- Network swapping
  - InfiniSwap
- Non-coherent distributed memory
  - VMware
- How to communicate across nodes?
  At what level of transparency?

Message Passing

- New programming languages
  - Use message passing instead of shared memory to do thread communication
  - Golang channel, Erlang actors, Akka library
- New application development practices
  - Use multiple processes instead of multithreading
  - Use message passing instead of shared memory
  - Nginx, Apache Web Server, Node.js

Splitting Containers with Message Passing

- Splitting a container across physical machines
  - Both processes and memory
  - Elastic, good resource utilization

- Use message passing for both inter- and intra-process communication
  - No coherence traffic
  - Easy to track and optimize inter-node communication

Challenges in Splitting Containers

- Performance overhead of crossing network
- Performance imbalance
- Management of split resources
- QoS
- Failure handling

Conclusion

- Splitting computation and memory not a new idea
- But new application and network properties
- Splitting container across physical machines
- Use message passing for all inter- and intra-process communication
- More challenges and opportunities

Thank you
Questions?

WukLab

wuklab.io

PURDUE UNIVERSITY