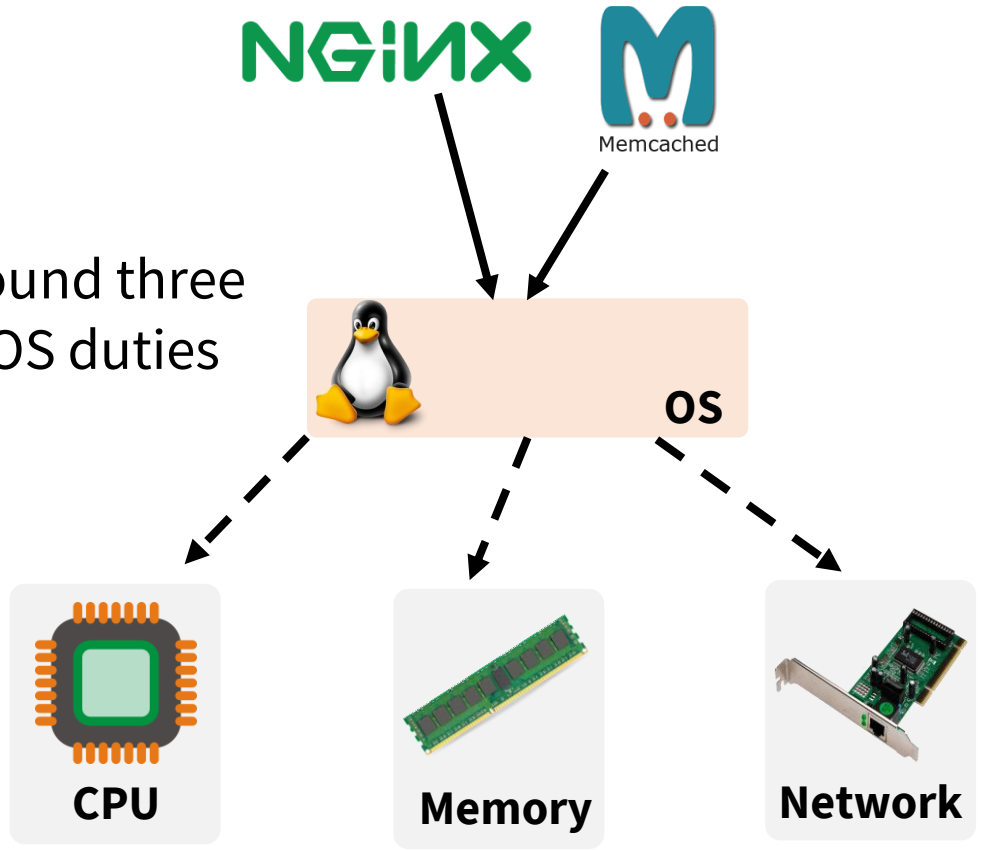
OS and Hardware

Adil Ahmad

PURDUE
UNIVERSITY

Categorizing the papers at OSDI and ATC

Centered around three important OS duties



And new ways to improve the current hardware

A. Process scheduling

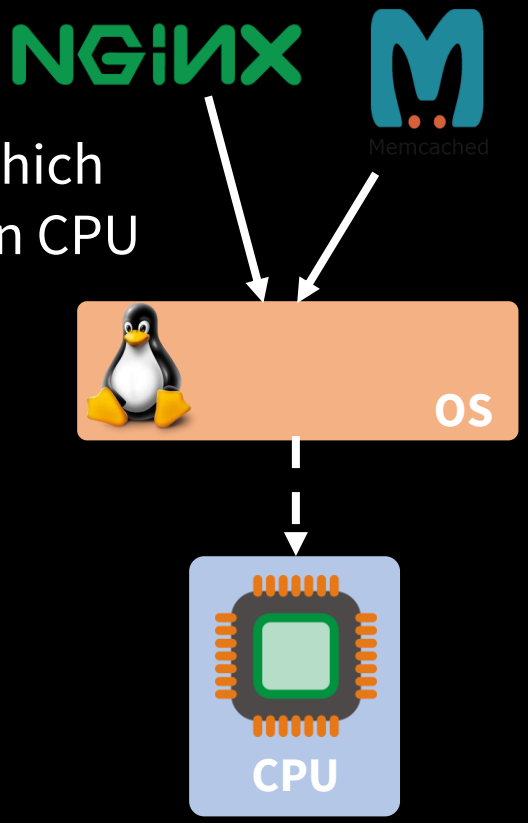
B. Memory management

C. Network management

D. Novel hardware designs

A. Process scheduling

OS decides which process runs on CPU



Two relevant concepts:

Fairness: each process should be allowed to use the CPU equally

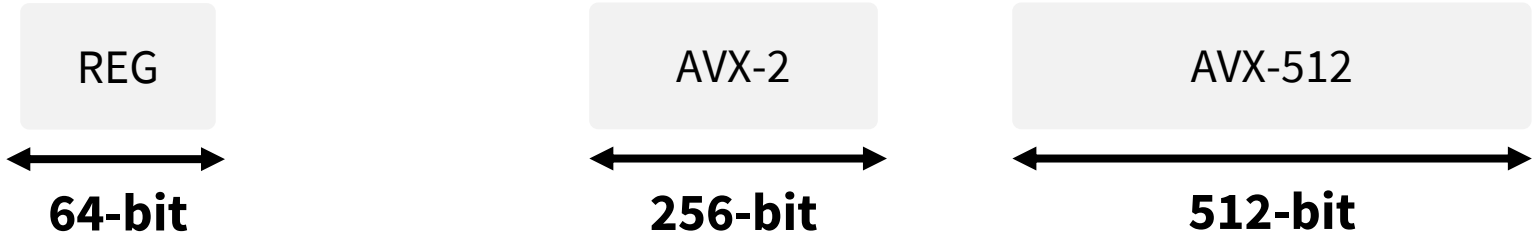
Latency: each process should complete as soon as possible

Paper #1

“Fair Scheduling for AVX2 and AVX-512 Workloads”
USENIX ATC 2021

Identify that AVX causes scheduling unfairness and compensate the affected processes

Background: AVX instructions

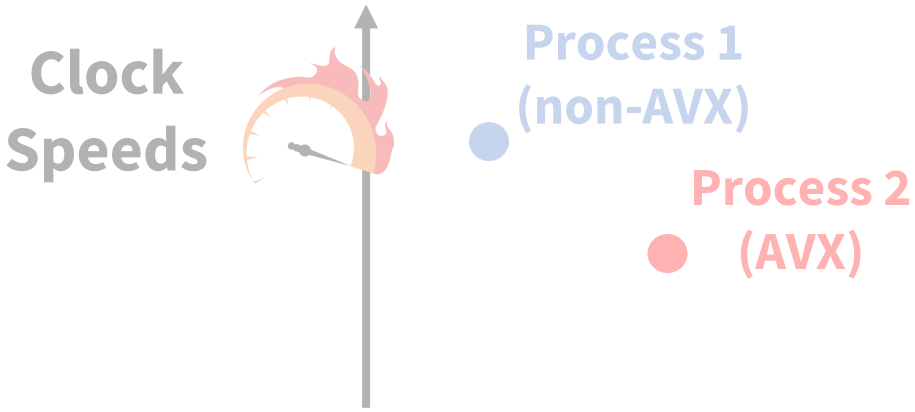


Non-AVX process data in 64-bit registers

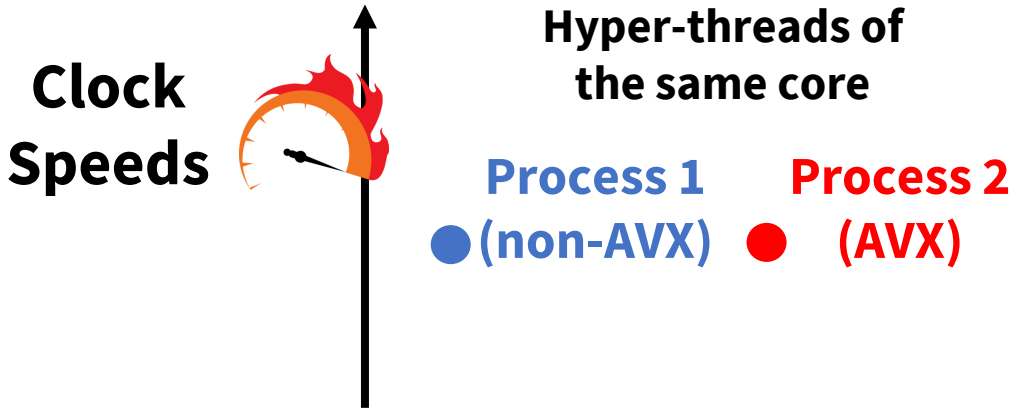
AVX can process data in 256 or 512-bit wide registers at a time

Better performance for certain applications!

Problem: unfairness from AVX instructions



AVX instructions force a CPU core into a lower clock speed



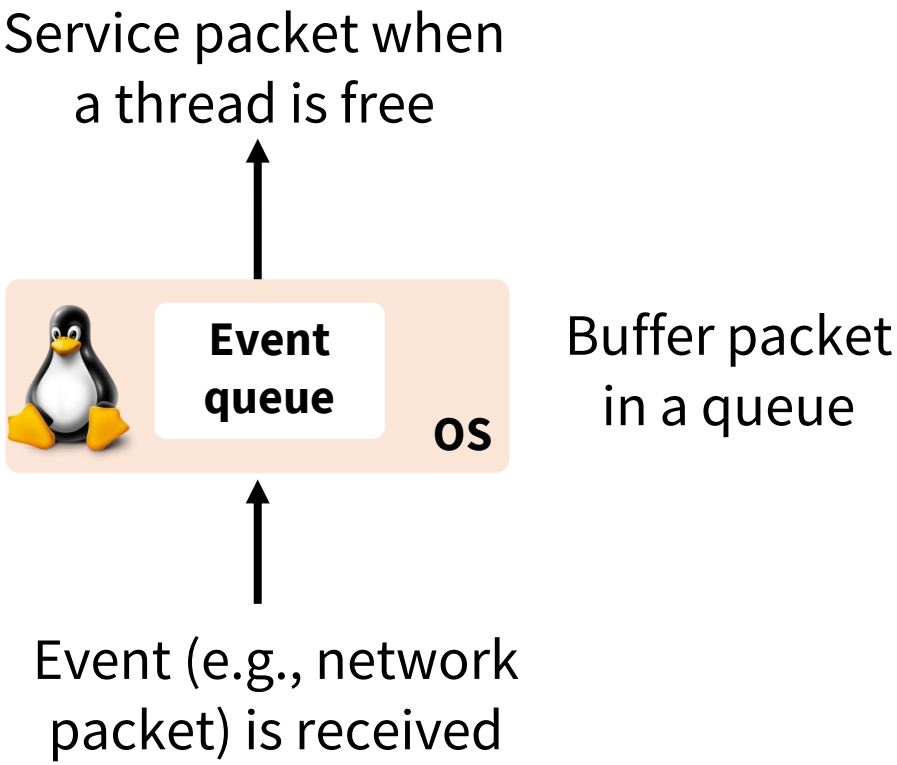
Other processes executing on the same processor core are affected!

Paper #2

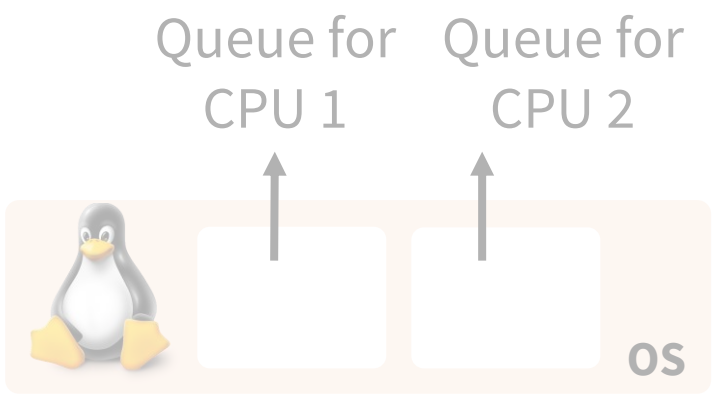
“SKQ: Event Scheduling for Optimizing Tail Latency in a Traditional OS Kernel”
USENIX ATC 2021

Design a scalable hybrid event queue
to reduce application latency

Background: event queues

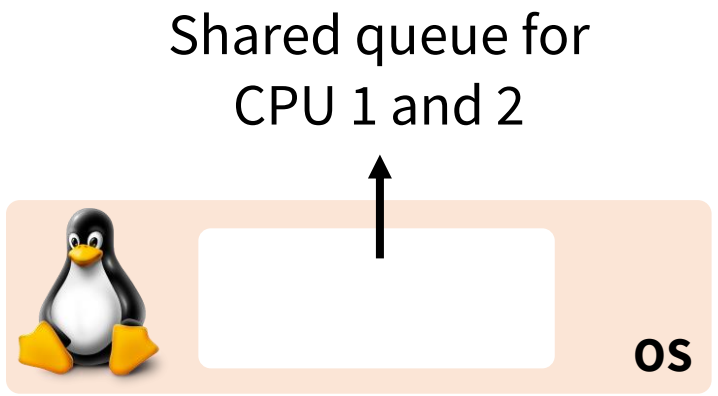


Problem: latency in current event queues



1:1 model

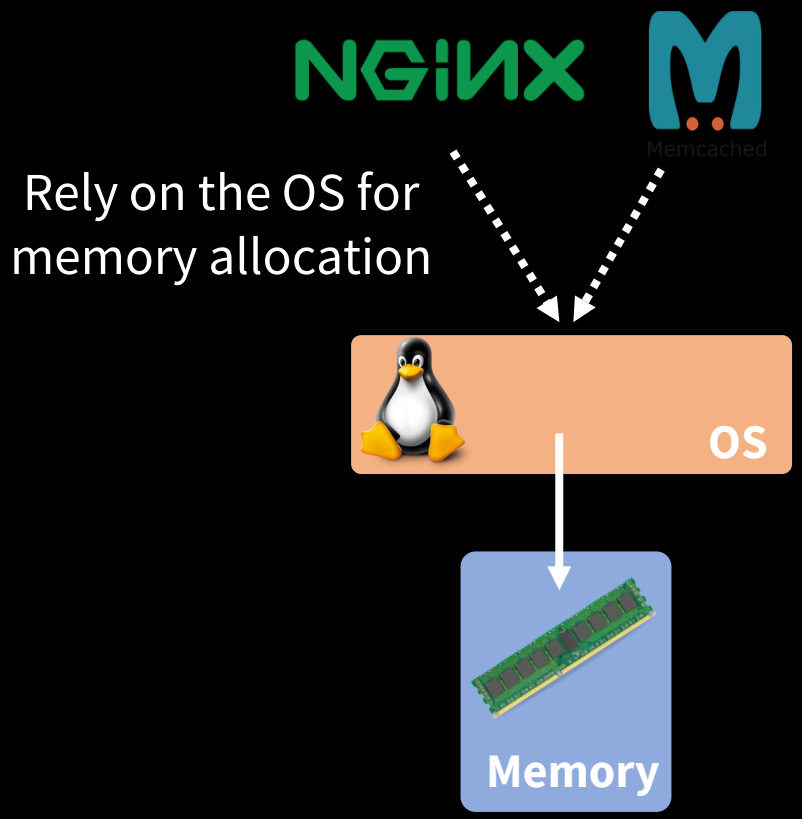
Under-utilized CPUs because events cannot be migrated easily!



1:N model

Expensive synchronization between CPUs on each event!

B. Memory management



Two relevant concepts:

Memory conservation: support many programs with limited memory

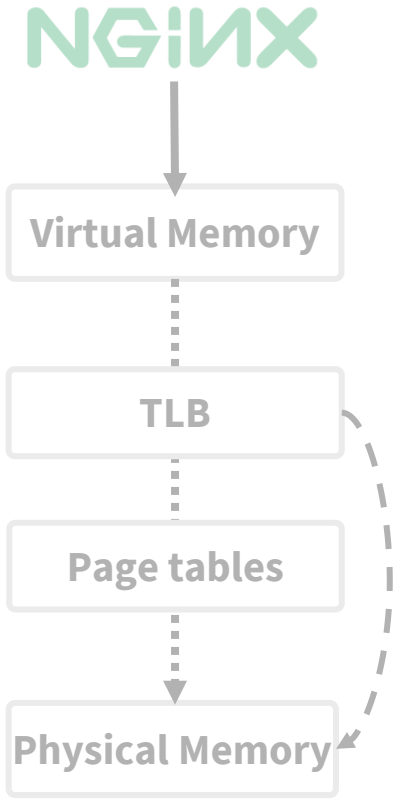
Maximum performance: reduce the performance impact of memory accesses

Paper #1

“Beyond Malloc Efficiency to Fleet Efficiency: A Hugepage-aware Memory Allocator” USENIX OSDI 2021

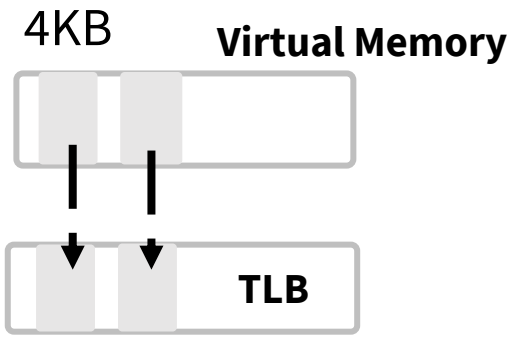
Create a memory allocator which helps conserve memory for hugepages

Background: translation and hugepages



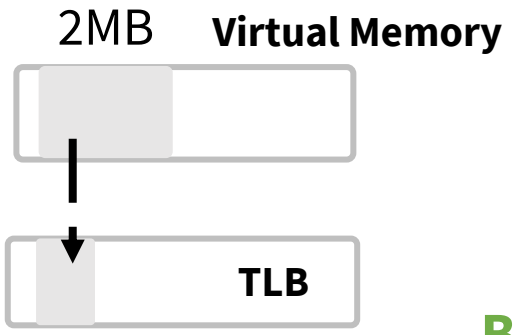
Fast translations using TLB

Slow translations using page tables



Normal pages

Lots of TLB entries needed!



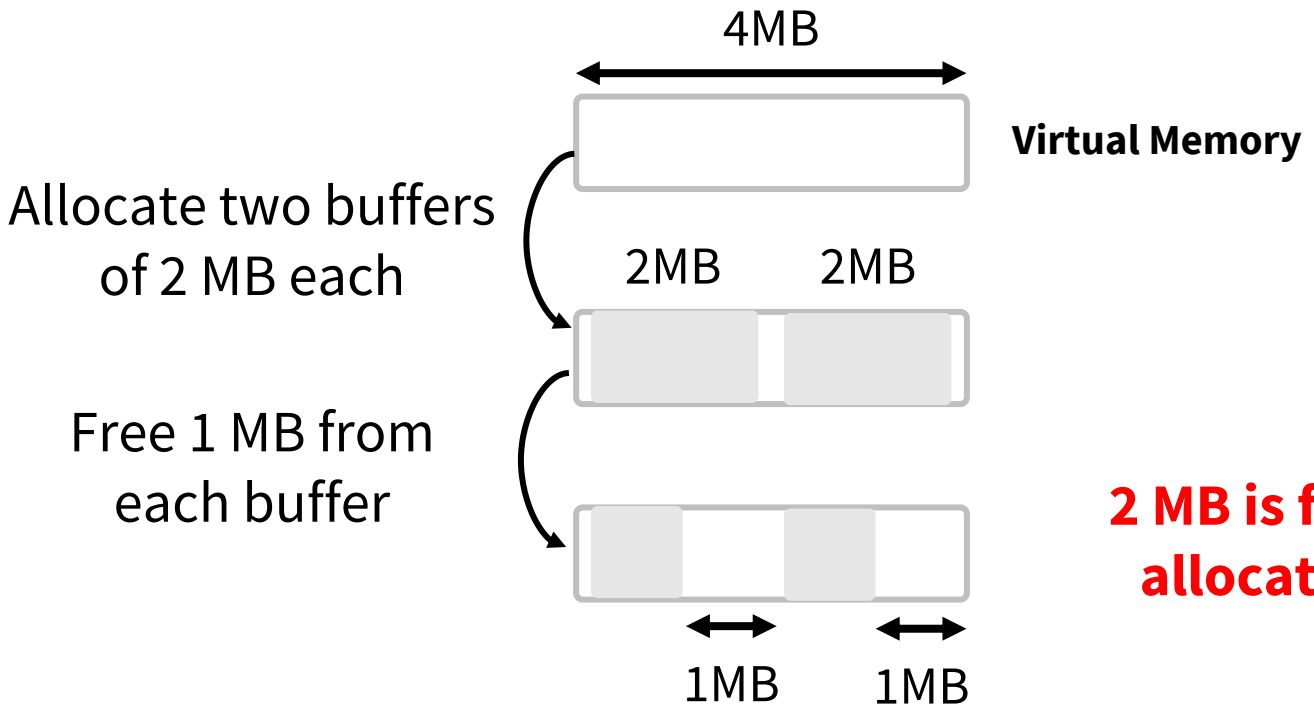
Huge pages

Fewer TLB entries needed!

Better performance!

"Beyond Malloc Efficiency to Fleet Efficiency: A Hugepage-aware Memory Allocator" USENIX OSDI 2021

Problem: fragmentation from hugepages



2 MB is free but can only allocate 1 MB buffers!

Paper #2

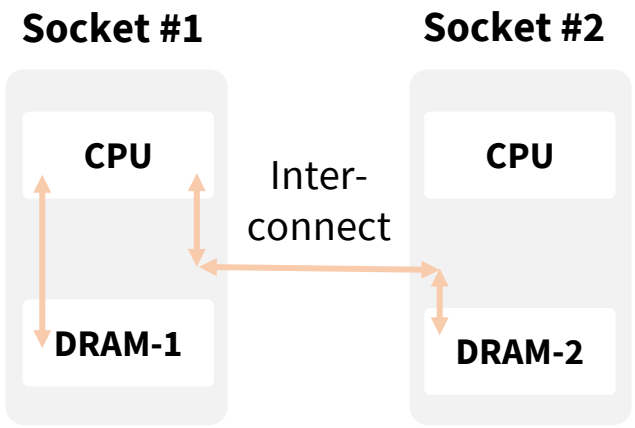
“NrOS: Effective Replication and Sharing in an Operating System”
USENIX OSDI 2021

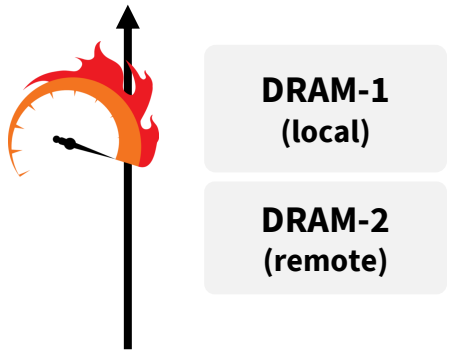
Design an OS kernel which is optimized for server architectures (i.e., NUMA)

Background: non-uniform memory access

Can access both local and remote DRAM

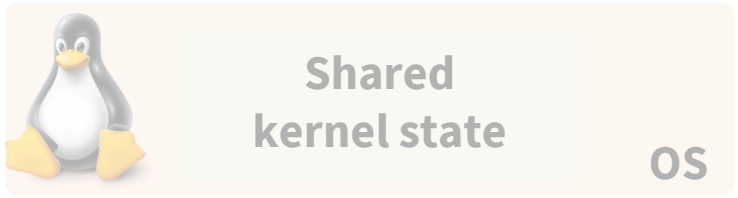
Memory access speed depends on locality



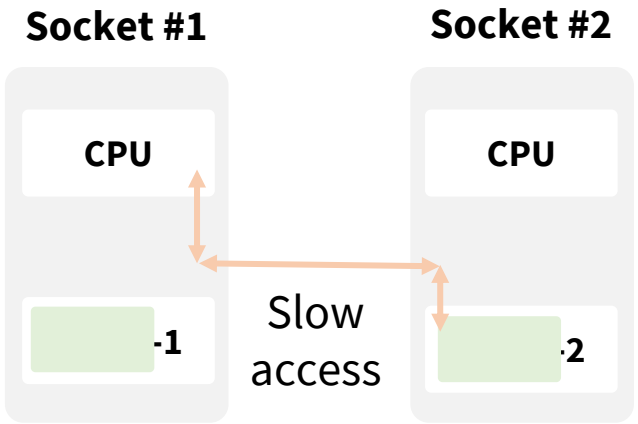


Non-uniform memory access (NUMA)

Problem: NUMA slows monolithic kernels



A shared kernel state is accessed by all CPUs



Frequent accesses to remote DRAM harm performance!

Paper #3

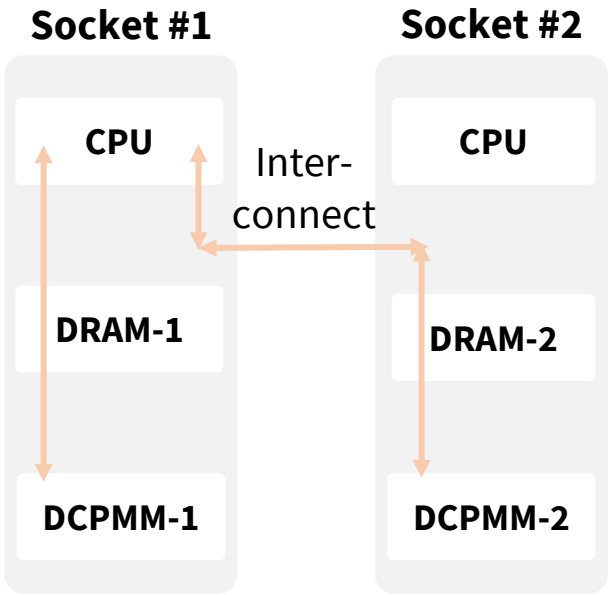
“Exploring the Design Space of Page Management for Multi-Tiered Memory Systems”
USENIX ATC 2021

Design a memory management scheme
optimized for server memory architectures

Background: multi-tiered memory (MTM)

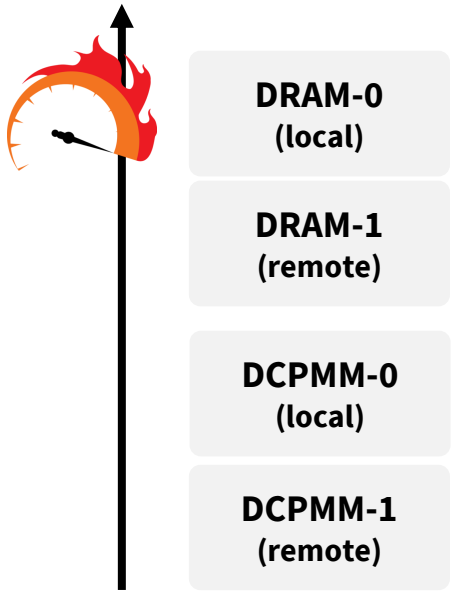
DRAM is a “fast” memory tier

DCPMM is a “slow” memory tier



Multi-tiered memory with NUMA

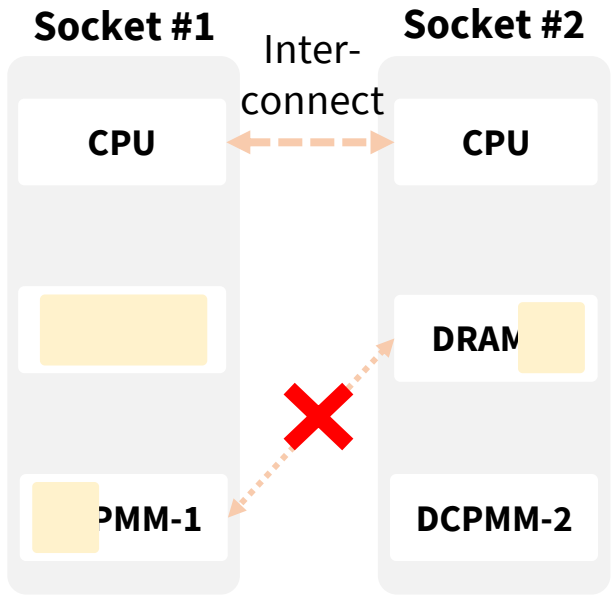
Memory access speeds depend on memory tier and locality



Problem: inefficient MTM usage

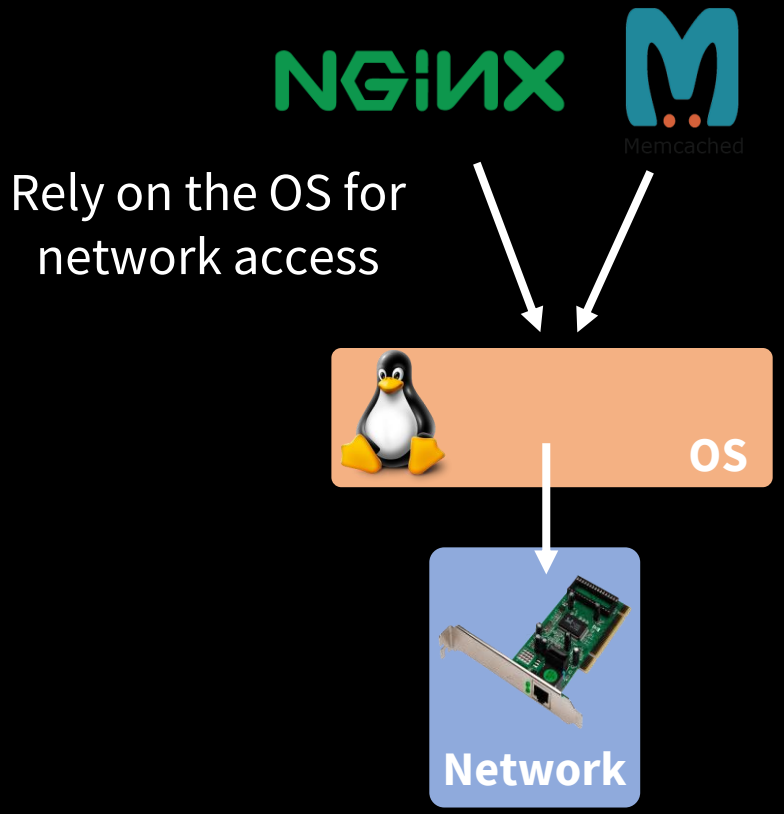
Local DRAM is full of data

OS keeps data in local DCPMM to preserve locality



There is space in remote DRAM, which is faster!

C. Network management



Two concepts are relevant:

1. Network flows are **short** (e.g., web requests) or **long** (e.g., database snapshots)

2. **Short** network flows are more **latency-sensitive** than long ones

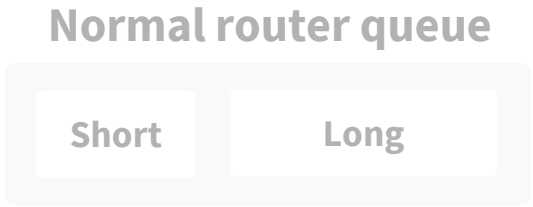
Paper #1

“A Linux Kernel Implementation of the Homa Transport Protocol”
USENIX ATC 2021

Implements Homa, a transport protocol that prioritizes short flows to reduce latencies

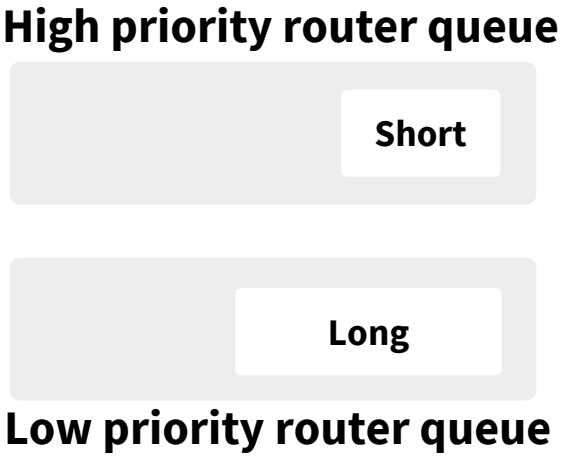
Background: Homa transport protocol

TCP:
Sender does not control how packets are handled by network



Latency-sensitive short flows could be delayed!

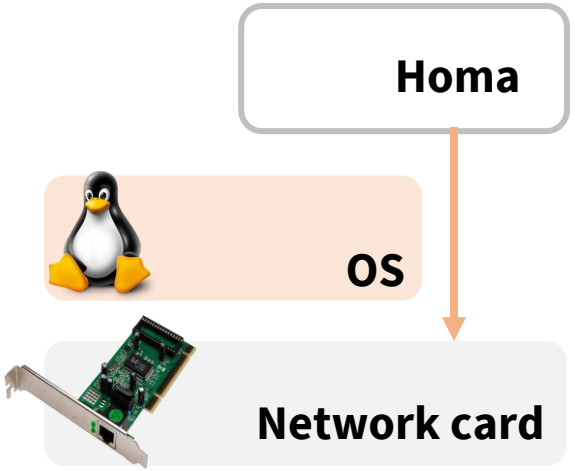
Homa:
Sender configures packets to use priority queues in routers



Short flows are prioritized, and latency is reduced!

Problem: Homa is not tested practically

Simulations through software network simulators



User-level implementation with kernel-bypass

D: Novel hardware designs

Improve **security** of
remote computation

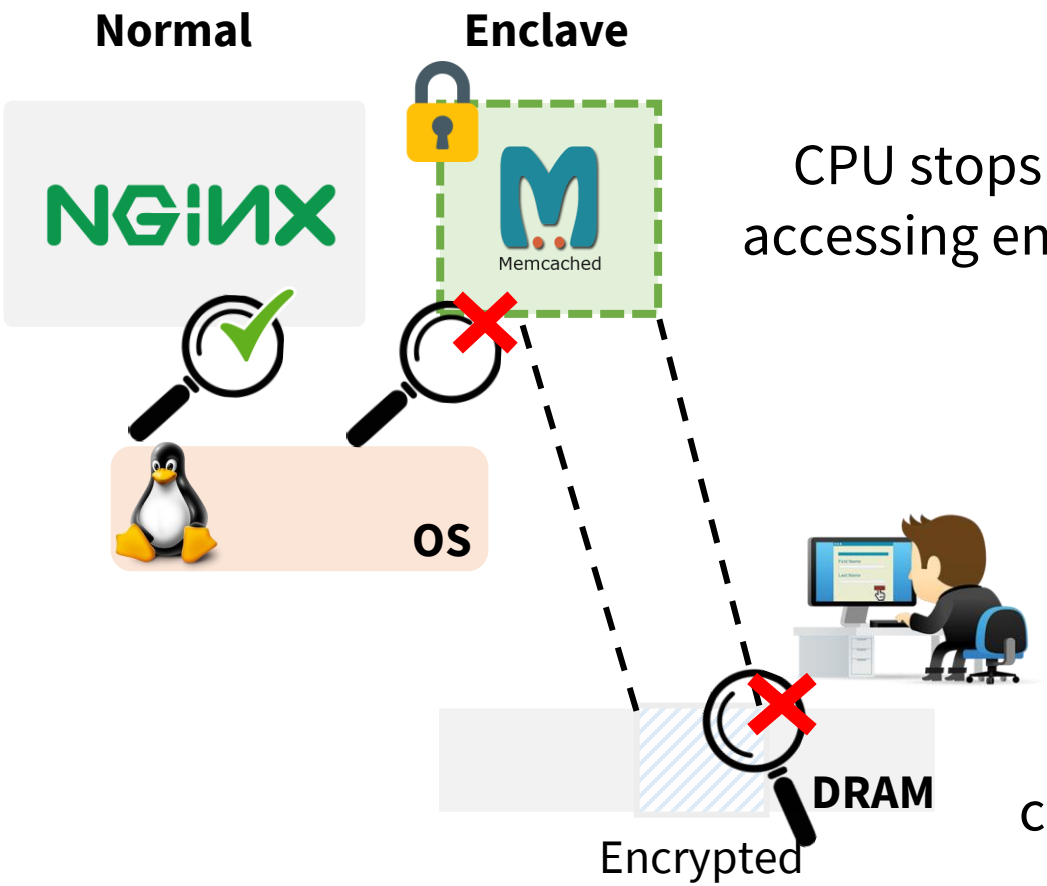
Improve **performance** in
emerging real-world scenarios

Paper #1

“Scalable Memory Protection in the Penglai Enclave”
USENIX OSDI 2021

A new trusted hardware that is more scalable and robust than previous solutions

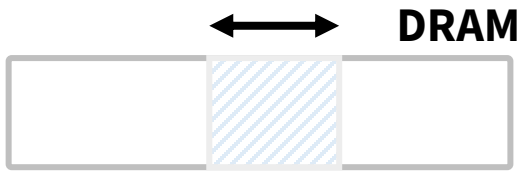
Background: enclaves



CPU stops the OS from accessing enclave memory

Even physical attackers cannot leak enclave memory

Problem: existing enclaves do not scale



Limited enclave memory (e.g., 128-256 MB)

Slow startup speeds (e.g., many seconds)

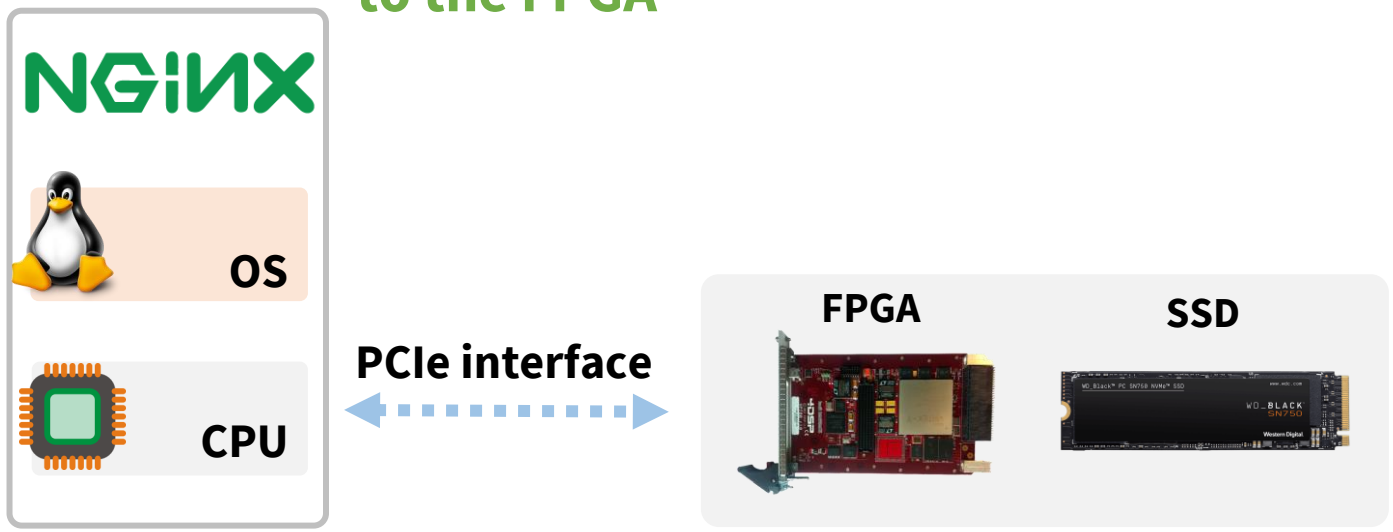
Paper #2

“A Fast and Flexible Hardware-based Virtualization Mechanism for Computational Storage Devices” USENIX ATC 2021

A new hardware to virtualize computational storage devices without relying on slow software

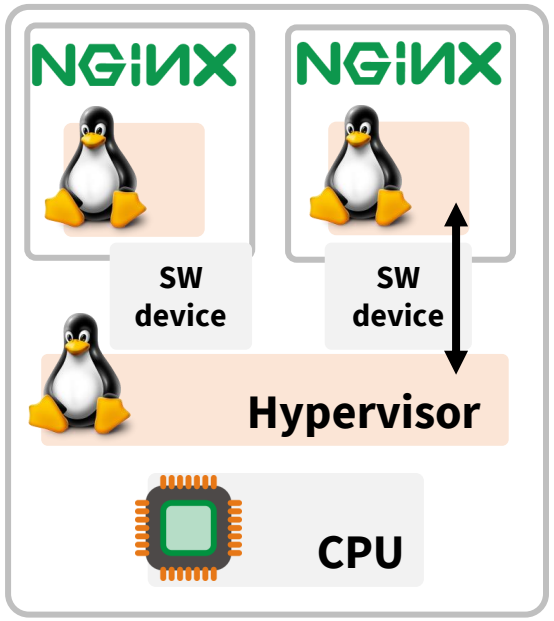
Background: computational storage

Offload computation
to the FPGA



Reduces data moved
through slow PCIe

Problem: CSD virtualization is inefficient

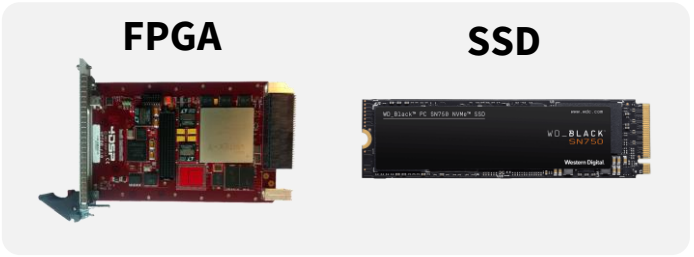


Main machine

Must rely on slow software interfaces!

PCIe interface

CSDs do not support hardware virtualization



Computational storage device

Paper #3

“The nanoPU: A Nanosecond Network Stack for Datacenters”
USENIX OSDI 2021

A new hardware to realize nanosecond response times for very short-lived datacenter requests

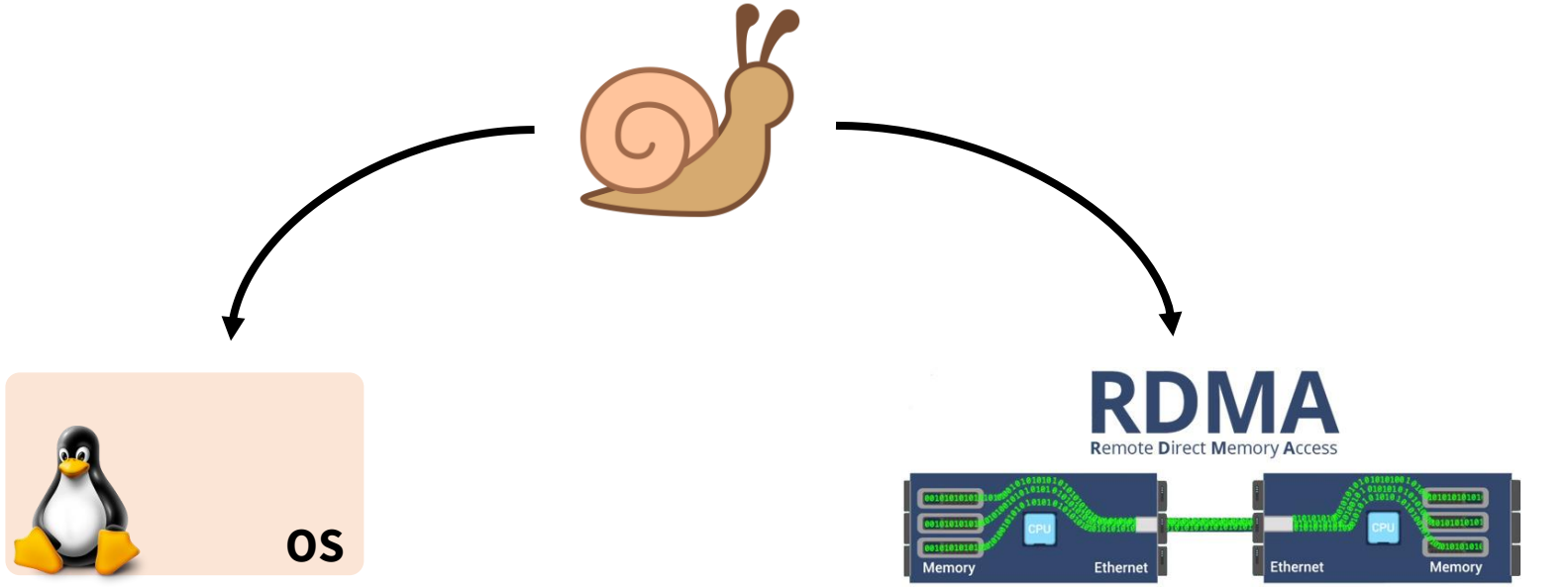
Background: remote procedure calls

Can call a function (or procedure) on a remote machine as if it was a local function

Few nano-second (ns) – 1 micro-second (us)

RPCs in datacenters are becoming more and more short-lived

Problem: existing network stacks are slow



Software stacks (e.g., Linux) take ~100s of microseconds

Hardware stacks (e.g., RDMA) take 1-2 microseconds

Conclusion

OS and hardware

Thursday, July 15 @ 7 AM PDT

My tail never has any latency

Friday, July 16 @ 8.30 AM PDT