RMAI: Rethinking Memory for AI

In-Kernel Remote Shared Memory as a Software Alternative to CXL

Amir Noohi (amir.noohi@ed.ac.uk) **University of Edinburgh**
Mostafa Derispour(m.deris@ec.iut.ac.ir) Isfahan University of Technology
Antonio Barbalace(antonio.barbalace@ed.ac.uk) University of Edinburgh

EuroMLSys 2025, Rotterdam, Netherland

Introduction

AI Models are Rapidly Growing:

- Modern models such as Switch Transformer, GLaM, and M6-T exceed **trillions of parameters**
- Model parameters often exceed single-node memory capacity during inference

Existing Solutions

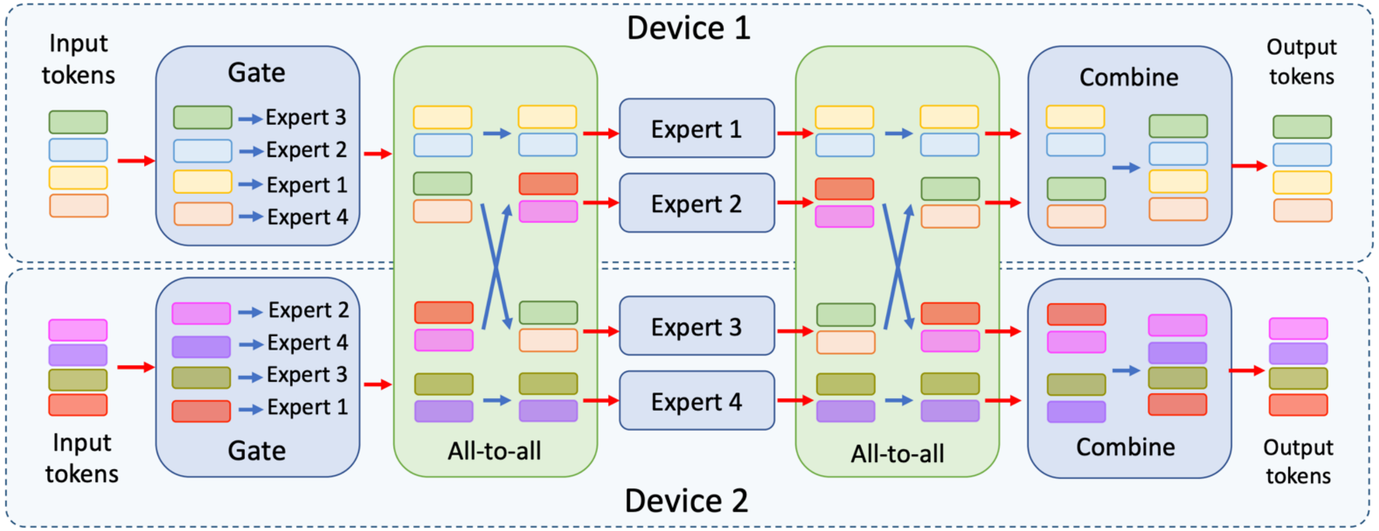
- ✓ Compute Express Link (CXL) for memory expansion
- ✓ Disk-based offloading
- ✓ Custom memory management techniques

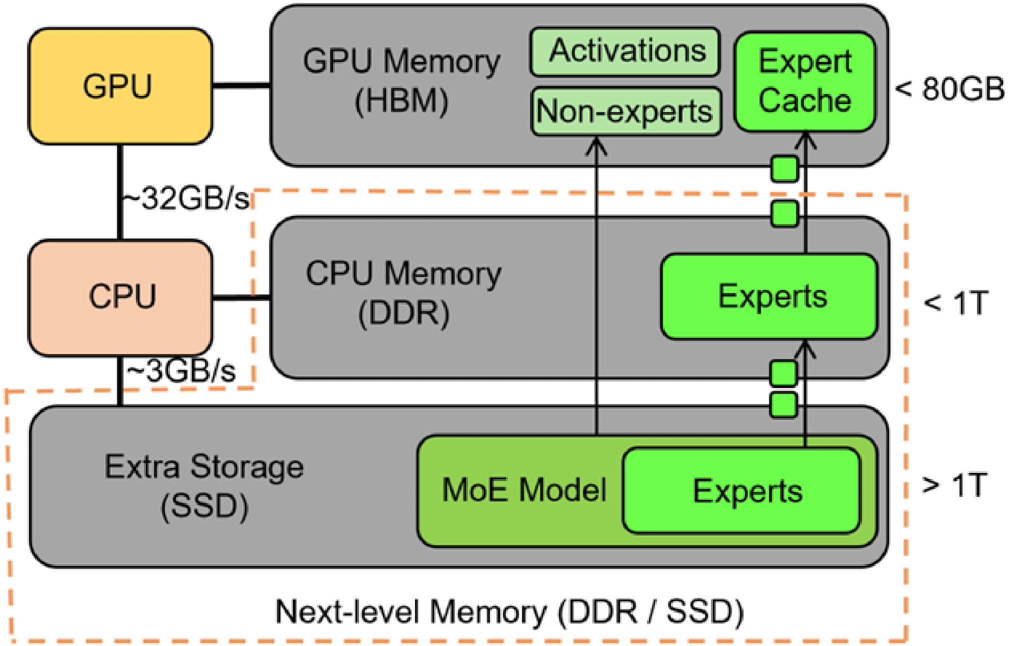
Evolution of Modern Datacenters

- Shift towards **disaggregated architectures** to address resource imbalances
- Separation of compute and memory nodes for better utilisation
- Advances in modern CPUs in datacenters(e.g., CPUs with accelerators: Intel AMX, ARM SME)

Background

- Experts are selected based on Input tokens on demand
- few experts get selected ➔ sparse memory access pattern
- The model is too big to fit ➔ offload it to disk or CXL devices





Motivation – Challenges with Existing Solutions

Limitations of Disk-Based Offloading

- Severe latency (milliseconds-scale) not suitable for real-time inference workloads.
- Low effective bandwidth (limited to ~7GB/s with NVMe storage).

Challenges with CXL (Hardware-Based) Solutions

- High capital investment and operational costs.
- Limited scalability due to hardware constraints:
  - Restricted to within-rack deployments (limited physical distances).
  - Multi-hop switches add significant latency and complexity.
- Limited availability and slow adoption rate in datacenters.

Resource Underutilization in Datacenters

- Many datacenters report low memory utilization (40%-60%).
- Underutilized memory resources on nodes not actively engaged in high-performance computation.

Table 1: Comparison of Memory and Interconnect Technologies

Technology	Cost	Availability in Data Centers	Latency	Bandwidth
DDR4 RAM	$6.20–$12.40 per GB	Widespread	80–100 ns	208 Gb/s
CXL 1.1 Memory	$1,860 per 128 GB	Limited	245–255 ns	136–208 Gb/s
RDMA	$62–$1,240 per 25–800 Gb/s	High	1–2 μs	25–800 Gbps
NVMe Storage	$496 per 1 TB	Widespread	92,000–537,000 ns	36 Gb/s

Motivation – OS level memory management overheads

Frequent Expert Switching in MoE

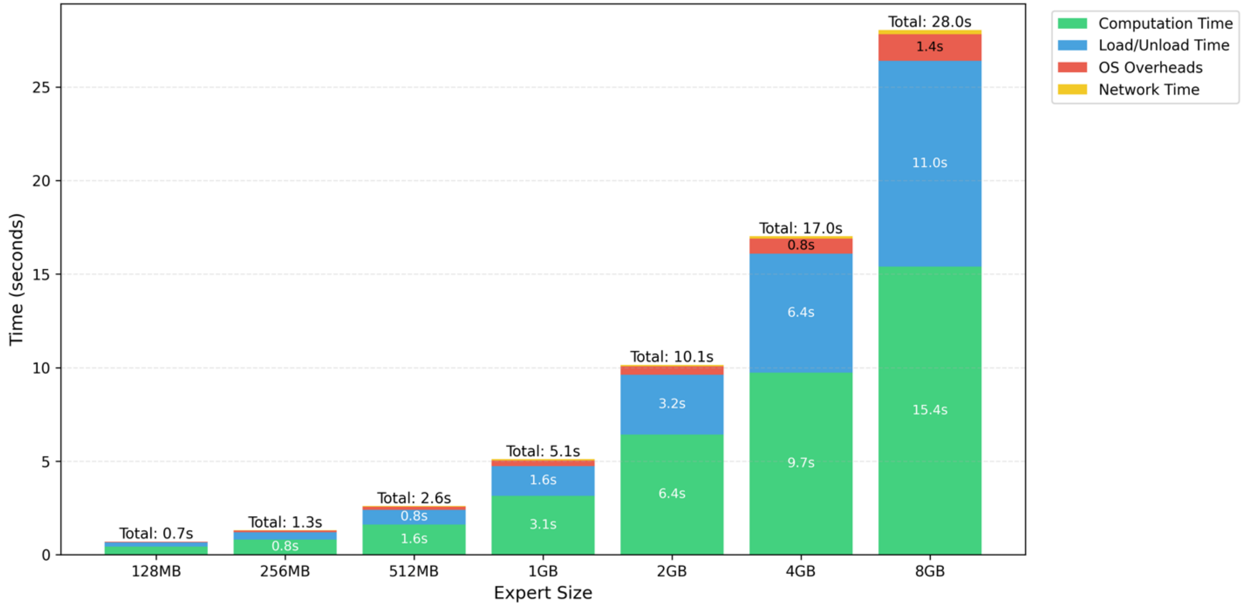
- Experts dynamically selected per inference request.
- load/unload experts When CPU memory is full each input request
- frequent expert migration into and out of memory (every ~50-100ms).

High Operating System Overhead

- Significant page fault overhead:
  - Loading a single 4GB expert (~1 million page faults using 4KB pages).
- Translation Lookaside Buffer (TLB) pressure and memory fragmentation degrade performance.

Empirical Performance Impact

- Load/unload grows with expert size
- OS overhead and expert migration



Solution: RMAI

RMAI: In-kernel remote Shared Memory via RDMA

- Transparent software alternative to hardware-based memory expansion(CXL)
- Leverages existing, underutilized memory across datacenter nodes efficiently

Key Innovations

- **Kernel-Level memory management:**
  - Automatic handling of data migration and memory deallocation
- **Symmetric Unified Virtual Address Space(PGAS-inspired):**
  - Transparent global memory view for seamless integration into existing AI workloads

Benefits

- Significantly reduces overhead compared to disk-based and CXL-based approaches
- Requires minimal modifications to existing AI inference applications

System Architecture

Architecture Components

- **Compute Nodes:**
  - Execute inference workloads, transparently access remote memory
- **Memory Node:**
  - Hold expert parameters, dynamically share memory via RDMA

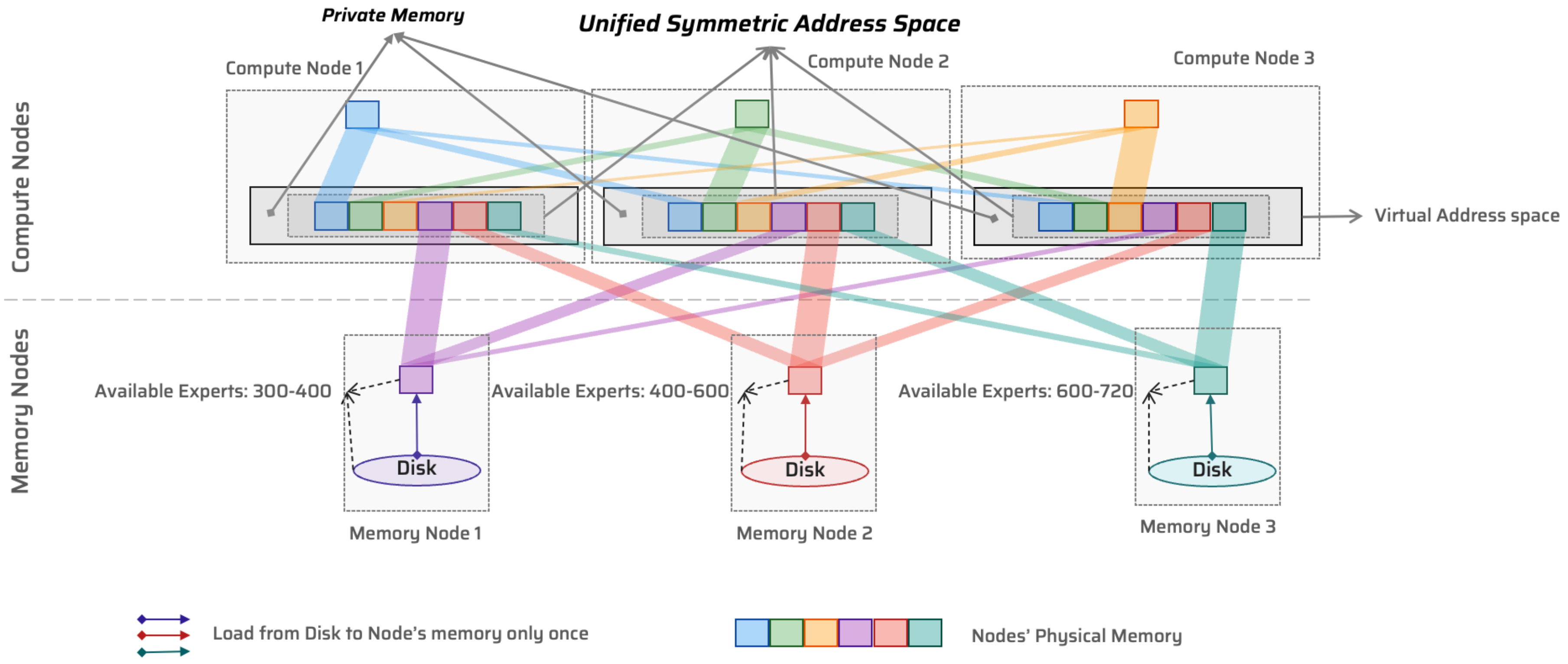
Efficient Memory Management

- **Virtual Memory Regions(VMRs):**
  - Coarse-grained, configurable regions reduce page faults and TLB overhead
- **Automatic Data Handling:**
  - Kernel transparently manages expert loading/unloading, data migration and caching
- **Page Deallocation Policy:**
  - Multiple copies of hot pages across both compute and memory nodes
  - LRU page deallocation mechanism to free up less demanding pages

Transparent integration

- No explicit user-space APIs or modifications needed
- Expert parameters transparently fetched via page faults using standard memory mapping(mmap)

System Design

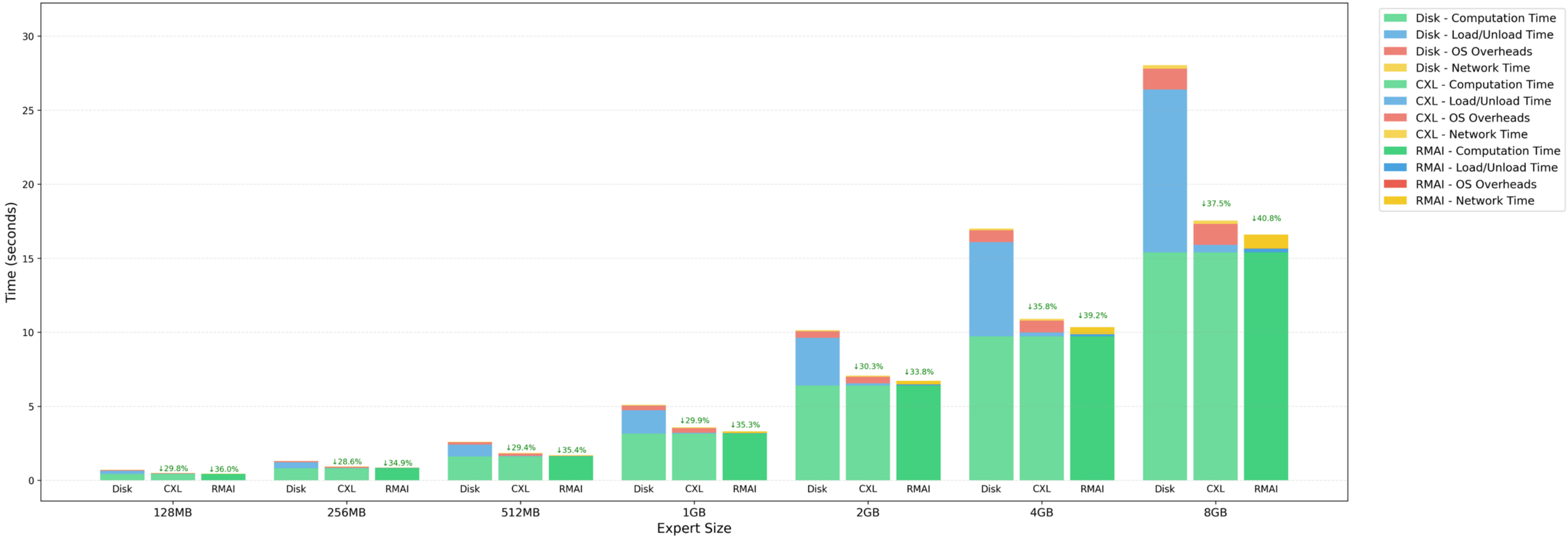


Evalution

Evaluation Setup:

Component	Specification
CPU	Intel Xeon Gold 5418Y (2.0GHz, 24 cores)
Memory	512GB DDR5 @ 4800 MT/s
Storage	465.8GB NVMe SSD (WDS500G1X0E)
Network	100GbE Mellanox ConnectX-6
CXL Device	Samsung CXL 1.1 DRAM Memory Expander (128GB)

Evalution (Continued)



- **Load/Unload time:** 95.4% improvement in CXL-based and 97.7% reduction in Disk-based
- **OS overheads:** 99% reduction
- **Scalability:**
  - small experts (128MB-1GB) 35.9%
  - medium experts (1GB-4GB) 39.2%
  - large experts (4GB-8GB) 40.7% reduction in total time.

Conculsion

- This work was a proof of concept for a software alternative for CXL in MoE workloads.
- Run faster than on CXL (up to 10%)
- Faster than the baseline (up to 45%) where the experts are on SSD
- Provides everyone with the ability to run MoE workloads using cheaper and more available hardware.

System	PGAS/DSM	Symmetric	Unified	Kernel-Level	Transparent	AI/Inference
INFINISWAP [20]	✗	✗	✗	✗	✓	✗
LEAP [30]	✗	✗	✗	✗	✓	✗
CFM [6]	✗	✗	✗	✗	✓(sch.)	✗
GMEM [45]	✗	✗	✗	✓	✓(dev.)	✗
HYDRA [25]	✗	✗	✗	✗	✓(part.)	✗
LEGOOS [37]	✗	✗	✗	✓(disagg.)	✓(part.)	✗
POPCORNOS [8]	✓(DSM)	✓	✓	✓	✓	✗
AIFM [14]	✗	✗	✗	✗	✗	✗
SAPS (ACTOR-PGAS) [33]	✓	✗	✗	✗	✗	✗
DRUST [29]	✓	✗	✓	✗	✓(lang.)	✗
RMAI (Ours)	✓	✓	✓	✓	✓	✓

THANKS FOR
YOUR ATTENTION

If there is any question, please reach out to me: amir.noohi@ed.ac.uk