Server Long Term RAM

Hui Sub (David) Shim and Katherine Mohr
Advised by Philip Levis

Problem

As DRAM's cost-per-bit has plateaued over the past decade, DRAM has become a major limiting factor in datacenter growth.

A new class of emerging memory technologies, known as **Long-term RAM (LtRAM)**, can be manufactured more cheaply and densely. This density comes with tradeoffs:

- Writes are **slower**
- Write granularity is **coarser** than a cache line
- Endurance is **bounded**

Device	Read lat.	Write lat.	Endurance	Write gran.	Cost/bit
DRAM	80 ns	80 ns	inf.	cache line	high
3D V-RRAM	100 ns	1 µs	10^6	page	low
3D FeFET	200 ns	10 µs	10^5	page	low
PCM	300 ns	1 µs	10^8	256 B	mid
STT-MRAM	20 ns	20 ns	10^{15}	word	high

Prior approaches to integrate LtRAM into servers have **exposed a DRAM-compatible interface** to the software, hiding LtRAM characteristics with **complex memory controllers**. However, hardware complexity **increases cost and performance variability**.

Optane Mechanism	Resulting Overhead	LtRAM Interface Fix
Cache-line Granularity Mismatch	**75%** write throughput drop, **2×** random-read latency	Cache-line reads, page writes; OS coalesces writes
Wear-Leveling	Opaque migration fires every ~3.4 MB written; **~60 µs tail read latency (160×)**	OS-side wear-leveling; hardware has no migration path
Address Indirection Table	**76–200 ns** added to every read	OS-side wear-leveling; no on-module map
Workload-Agnostic Design	70% read bandwidth drop under 50/50 R/W	Read-mostly by design; token allocator paces writes

hsshim@stanford.edu, kmohr@cs.stanford.edu

Solution

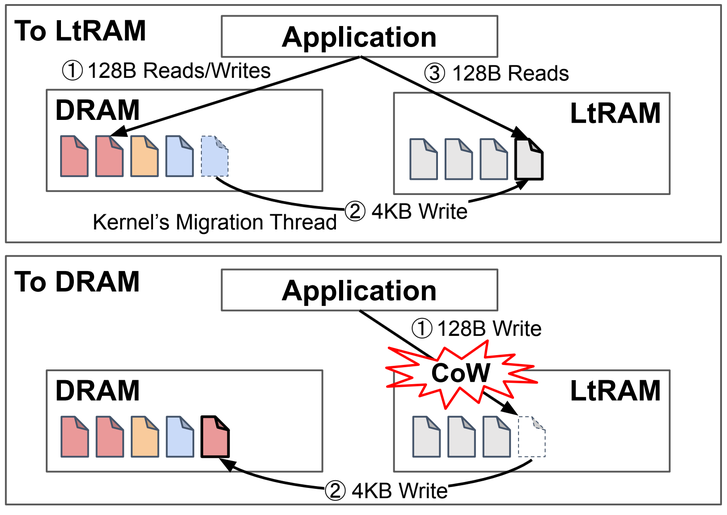
We introduce a novel HW/SW interface that enforces an invariant we call **Application Read-Only Memory (AROM).**

- Only the kernel may write to LtRAM when migrating pages from DRAM.
- LtRAM is **read-only to applications**.
- This design allows us to move LtRAM logic **from the on-DIMM controller to the OS**.

The on-DIMM controller supports **reads at cache-line granularity** and **writes at 4 KB page granularity**, and it exposes cell health metadata.

The **OS handles policy decisions** such as data placement, page migration, and wear-leveling.

Design

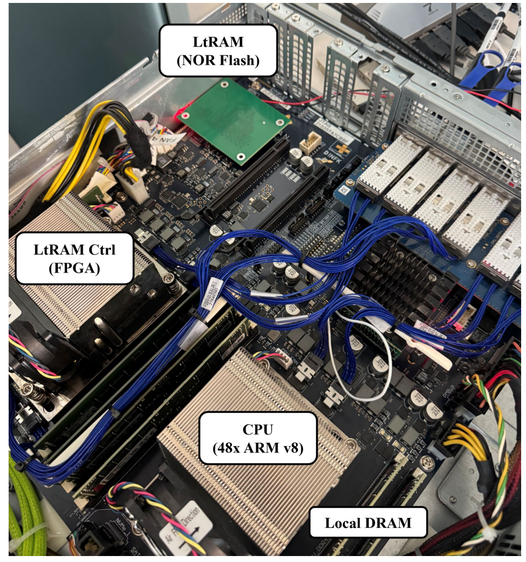


Challenges

- What is the **ideal LtRAM technology** to develop?
- What is the **best migration policy**?
- What is the **best wear-leveling strategy**?
- How can **out-of-memory faults be prevented** during write-heavy periods?

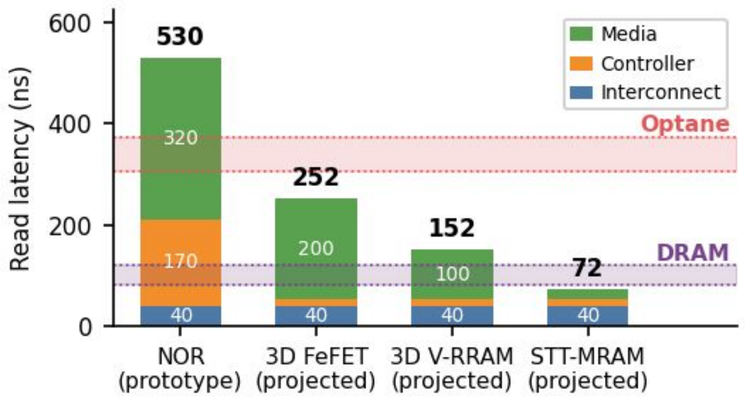
Evaluation

We prototype the interface on the **Enzian platform**, configured so the FPGA acts as the on-DIMM controller. **NOR flash** is chosen as the LtRAM technology due to its cost and availability.



Impact

While NOR flash is inherently slower than DRAM, **emerging LtRAM technologies approach DRAM read performance**, enabling mixed DRAM/LtRAM systems to match pure DRAM systems' performance.



Suppressing Contact-Doping-Induced Short-Channel Effects in Oxide Semiconductor Transistors *via* Contact Length Scaling

Chi-Hsin Huang*, Koustav Jana, Samantha Afra van Rijs and H.-S. Philip Wong*

Department of Electrical Engineering, Stanford University (*email: chihsinh@stanford.edu, hspwong@stanford.edu)

Highlight

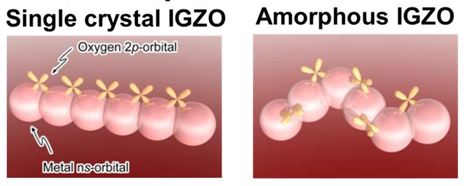
- Investigated the impact of channel length and contact length reduction in oxide semiconductor transistors.
- Demonstrated that shorter contact length effectively suppresses contact-doping-induced short-channel effects in oxide transistors.

Introduction & Motivation

Amorphous oxide semiconductor for BEOL transistor

Amorphous structure

Single crystal IGZO | Amorphous IGZO



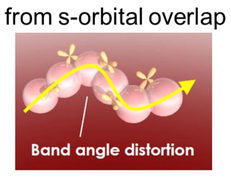
Electron mobility: ~10 $cm^2(Vs)^{-1}$ | ~10 $cm^2(Vs)^{-1}$

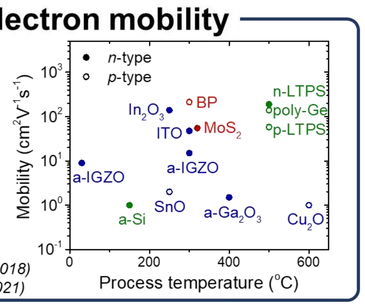
High ← Process temperature → Low

K. Nomura et al., Nature, 432, 488–492 (2004)

High electron mobility

~ 10–100 $cm^2/V.s$ from s-orbital overlap

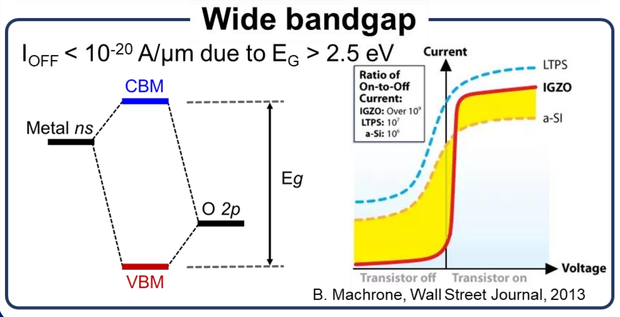




Nature Electronics 1, 442–450 (2018)
Appl. Phys. Lett. 118, 030501 (2021)

Wide bandgap

I_{OFF} < 10^{-20} A/μm due to E_G > 2.5 eV

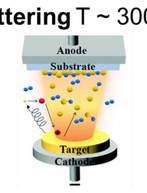


B. Machrone, Wall Street Journal, 2013

BEOL compatibility

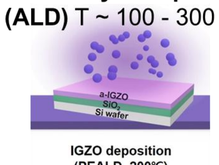
Process temperature < 400 °C

Sputtering T ~ 300 °C



Z. Zheng et al., J. Materials Chemistry C, 2017

Atomic Layer Deposition (ALD) T ~ 100 - 300 °C



IGZO deposition (PEALD, 200℃)

D. Kim et al., ACS AMI, 2023

Oxide semiconductor from display to memory and CMOS

Oxide thin-film transistors

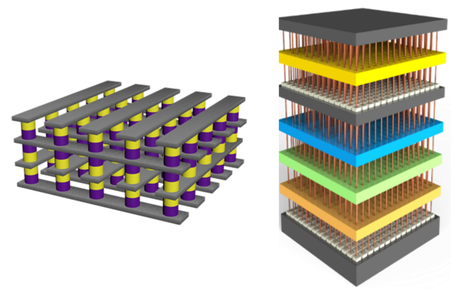
- Application: display (commercialized) IoTs, flexible electronics

Device size (channel length): 2~10 μm

Operation voltage: 3~ 15V

Oxide BEOL transistors

- Application: memory, CMOS



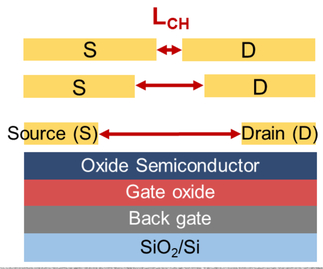
Device size (channel length): <100 nm

Operation voltage: <2V

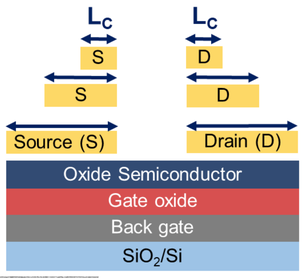
Si process compatibility

Oxide Semiconductor Transistor Scaling

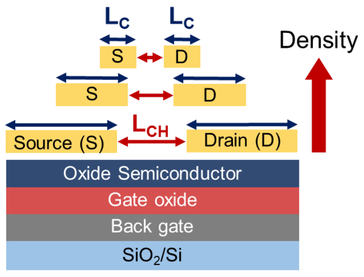
Channel Length (L_{CH}) Reduction

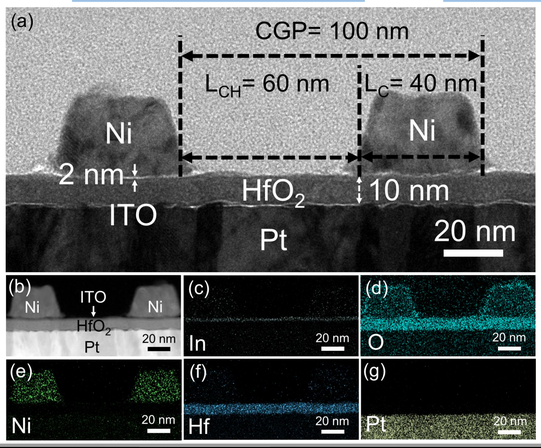


Contact Length (L_C) Reduction



Channel Length (L_{CH}) & Contact Length (L_C) Reduction





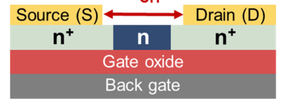
- Back-gate electrode Ti/Pt deposition
- HfO_2 Back-gate dielectric deposition (10 nm, ALD 200 °C)
- ITO channel deposition (2 nm, ALD)
- Channel patterning and etching
- S/D patterning and etching
- S/D contact (Ni) deposition (30nm, e-beam evaporation)

Process thermal budget ≤ 200 °C

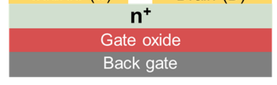
Contact-Doping-Induced Short-Channel Effects

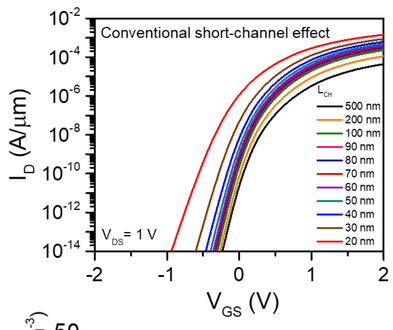
- Contact doping effect:
  (a) Hydrogen incorporation
  (b) Oxygen vacancies formation
  ➡ Donor-like defects

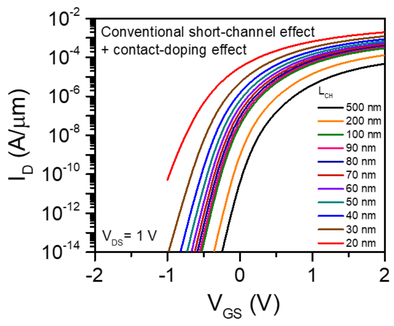
Long Channel Length (L_{CH})

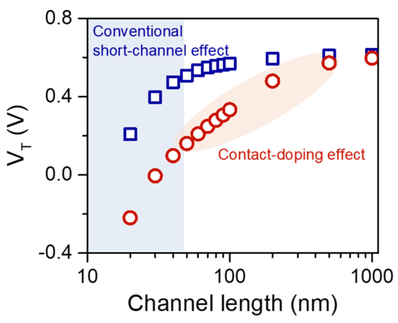


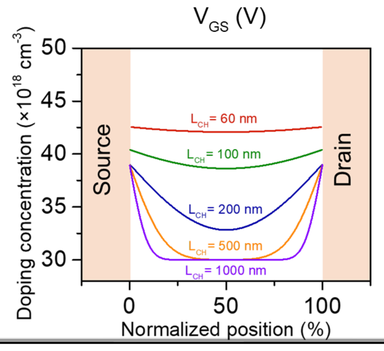
Short Channel Length (L_{CH})











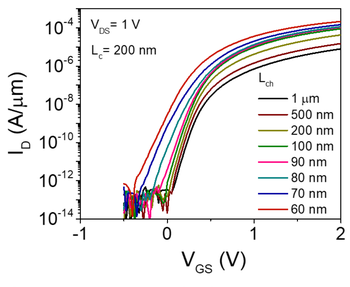
Lateral channel doping profile ($N_{CH}(x)$):

$$N_{CH}(x) = N_{PEAK}\left[erfc\left(\frac{x}{L_D}\right) + erfc\left(\frac{L_{CH}-x}{L_D}\right)\right]$$

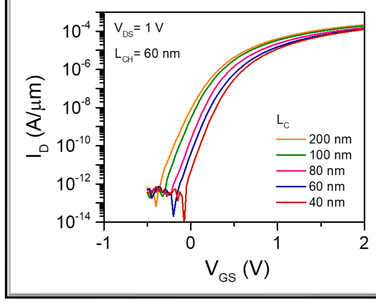
- N_{PEAK} is peak channel donor (V_O^{2+}/H^+) concentration at the S, D edges.
- L_D is the diffusion length of donor.

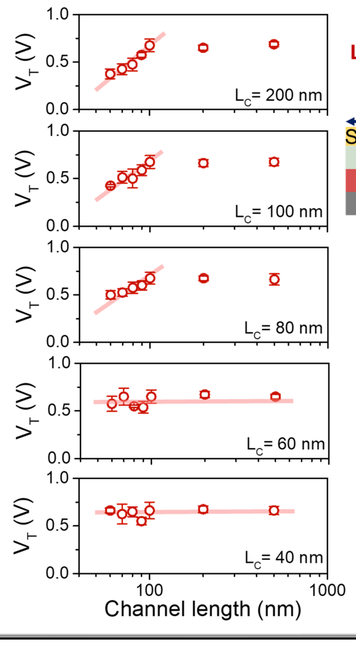
Channel & Contact Length Reduction in ITO FETs

Channel Length (L_{CH}) Reduction

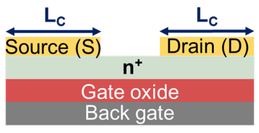


Contact Length (L_C) Reduction

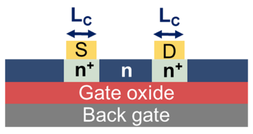


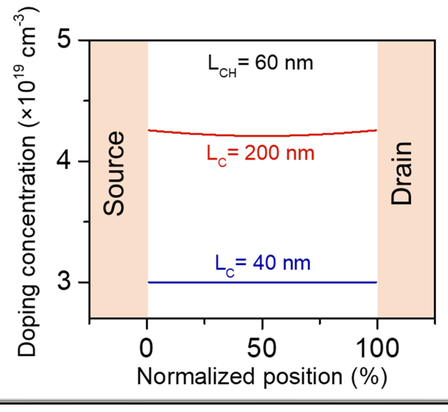


Long Contact Length (L_C) (200 nm)



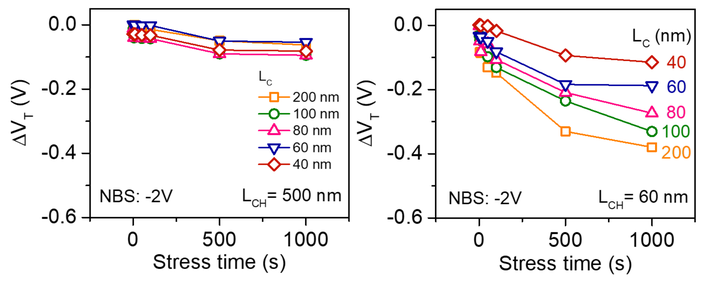
Short Contact Length (L_C) (40 nm)

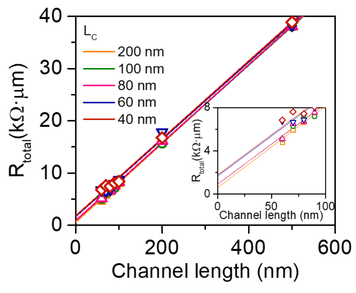




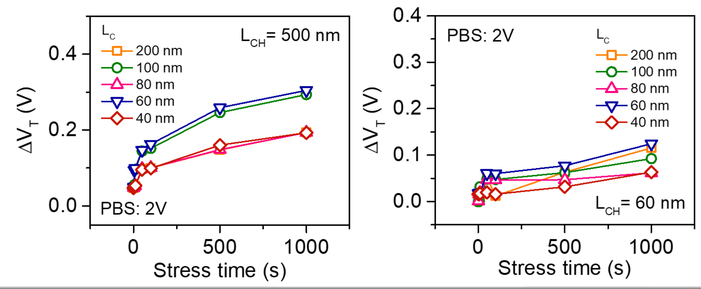
Bias-Stress Stability & Contact Resistance

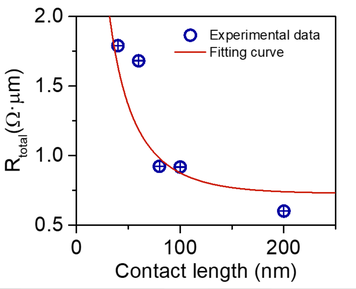
- Improved negative bias stress (NBS) stability by L_C reduction
- Contact resistance dependence on L_C





- Improved positive bias stress (PBS) stability by L_{CH} reduction





Conclusion

- Investigate the effects of reducing L_{CH} and L_C in ITO FETs.
- Shorter L_C effectively suppresses contact doping effect ➡ address V_T roll-off, SS degradation, and NBS instability.
- Demonstrate the enhancement-mode oxide FET with 80 nm CGP. (L_{CH}= 40 nm, L_C=40 nm)

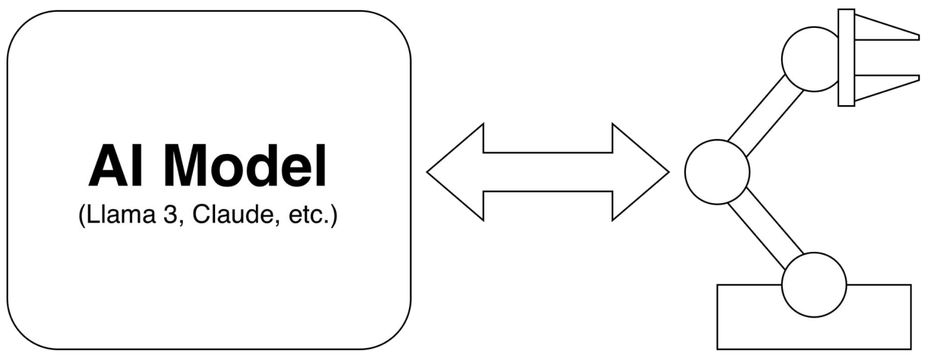
Reference: C.-H. Huang, K. Jana, S. A. van Rijs, and H.-S. Philip Wong, IEEE International Electron Devices Meeting (IEDM), 2025

µVLA: Designing a Multi-Chiplet 16nm SoP with Heterogeneous Memory on Package (HMoP) Enable Real-Time Physical AI on the Edge

Christian Kubicka, Thierry Tambe
Department of Electrical Engineering, Stanford University

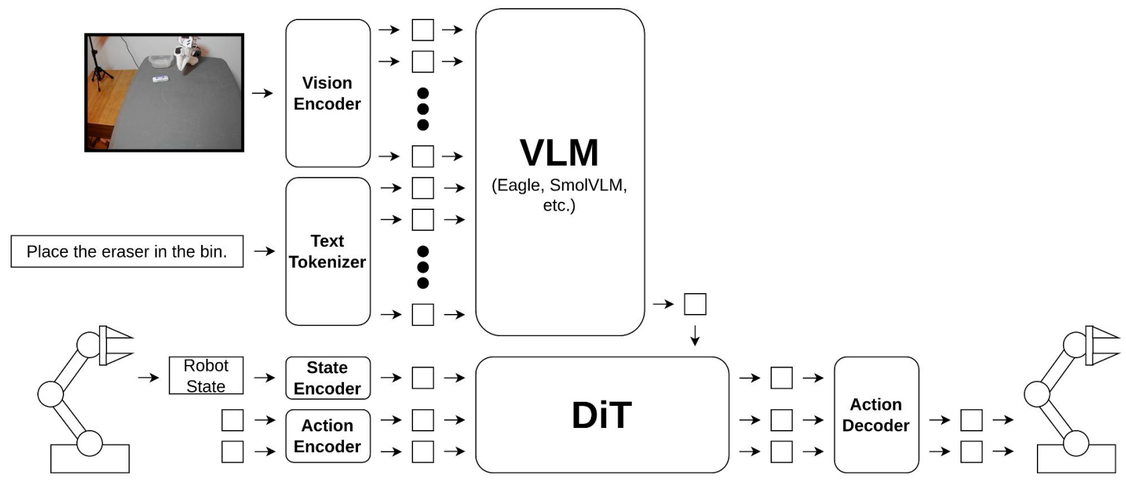
What is Physical AI?

- AI Models do well on a variety of tasks: coding, data retrieval, etc. However, by themselves, they lack the ability to interact with the real world.
- Physical AI is the concept of allowing AI Models to interact with real-world environments for the purpose of accomplishing some task.
- Most modern approaches rely on Visual Language Action (VLA) models to bridge this divide.



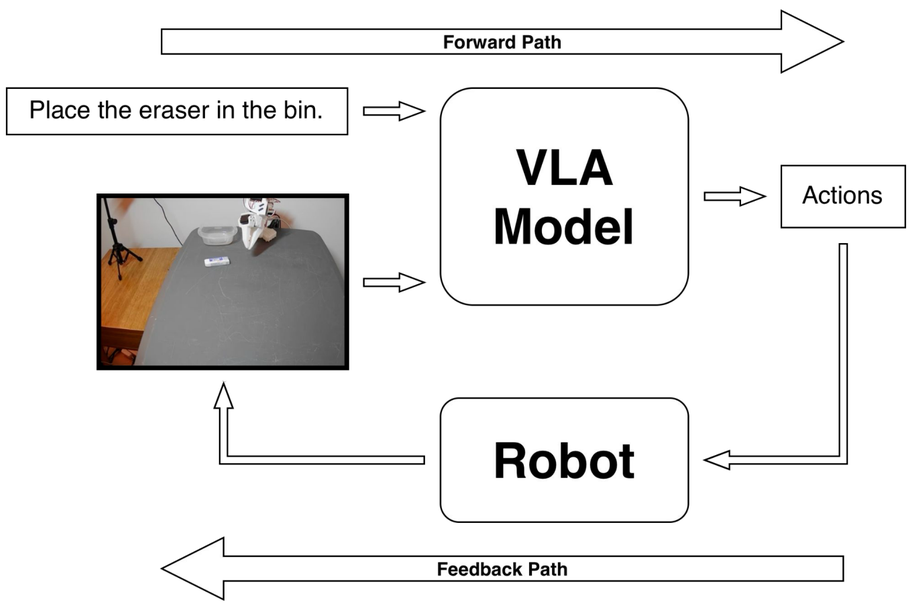
What is a Visual Language Action (VLA) model?

- VLA models use a pre-trained **Vision-Language-Model (VLM)** generate a single high-level action vector from a combined prompt and state.
- This high level action vector is then used as an input to a **Diffusion Transformer (DiT)** to extrapolate multiple new actions from higher-level vector.



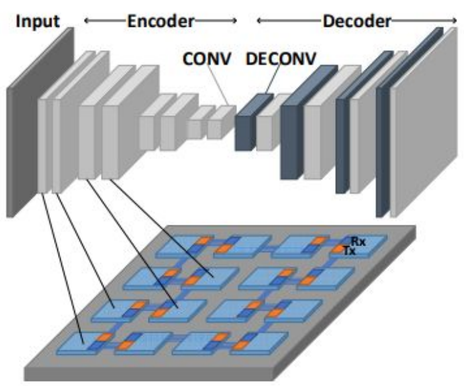
Why is this different from other inference tasks?

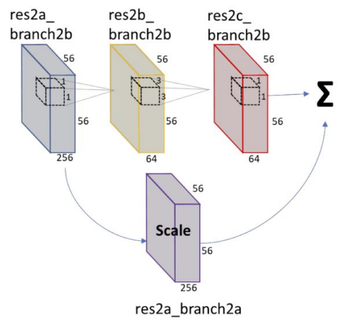
- This approach essentially implements **closed-loop control.**
- Therefore, single-batch latency is the primary design target in terms of performance, not throughput.

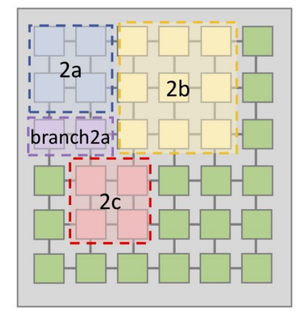


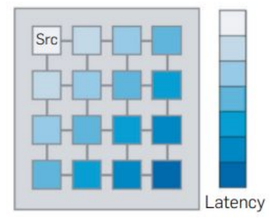
What do other chiplet works do?

- Most works ensure full device utilization through large batch sizes and spatial pipelining.
- **Current approaches fail under the extreme memory demands and the single batch nature of Physical AI algorithms.**









[1] https://ieeexplore.ieee.org/document/9830249
[2] https://dl.acm.org/doi/10.1145/3460227

How does µVLA overcome these problems?

- µVLA addresses the memory bandwidth/capacity through providing compute dies with direct access to two types of memory 1T1C DRAM and NOR Flash.
- µVLA enables high single-batch utilization through the use of a broadcast bus to distribute work to all the target dies at once.
- µVLA makes use of 3T GCRAM within the hub chiplet to store the transient data as it is gathered, reorganized and rebroadcast.

Memory Fit Predictor: Architectural Support For Page Placement in Heterogeneous Memories

Chun Deng, Hui Sub Shim
Advised by Philip Levis and Tsachy Weissman

Problem

DRAM is expensive and managing heterogeneous memory systems is hard due to a mismatch of abstractions.

OS policies lack visibility to memory-level hardware behavior, but governs memory allocation and placement:

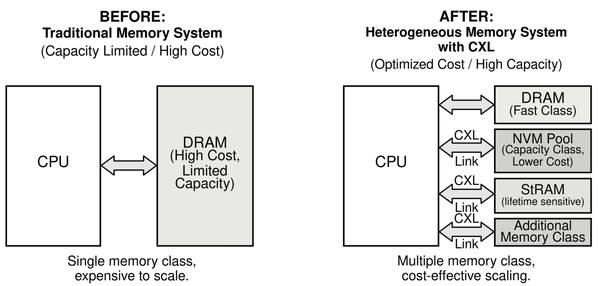
- Access bits are too coarse.
- PEBS overhead necessitates inaccurate sampling.
- Existing kernel solutions for heterogeneous memories rely on approximation models.

Memory technologies are coming with complex and subtle performance tradeoffs, and different systems can have different configuration of memories:

- Technology-specific policies break when memory systems change.

Memory	Read latency	Read bandwidth	Write Bandwidth
DDR5	114ns	38 GB/s	38 GB/s
Optane DC	172ns	32 GB/s	11 GB/s
CXL	239ns	22 GB/s	22 GB/s

- What is the right abstraction to support flexible heterogeneous memory management?



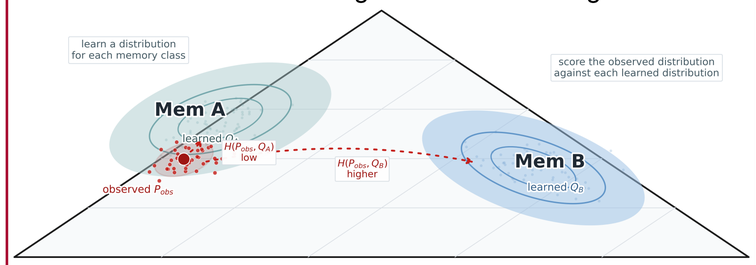
Goals

We aim to give the OS a learned hardware signal for which memory each page fits best - a simple abstraction called “memory fit,” computed efficiently by hardware and exposed to the kernel.

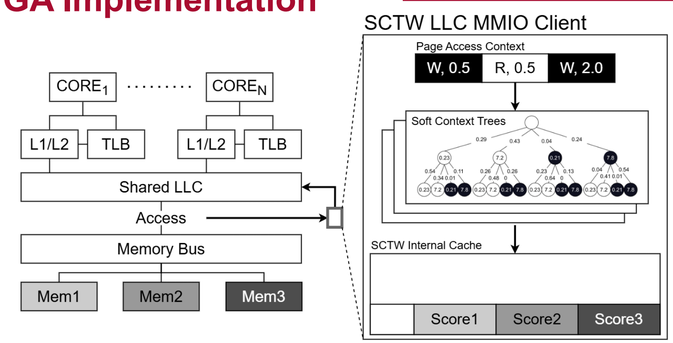
dengchun@stanford.edu

Solution

- We develop a novel technique called Soft Context Tree Weighting (SCTW), an online, hardware friendly estimator of memory access distributions inspired by compressors which learn the underlying source distribution.
- Memory fit is computed by measuring the cross-entropy between an access pattern and the learned soft context trees, measured in “bits” of information. Another way to think of this is how “surprising” a memory access is.
- We train our models using contrastive ranking loss with a



FPGA Implementation

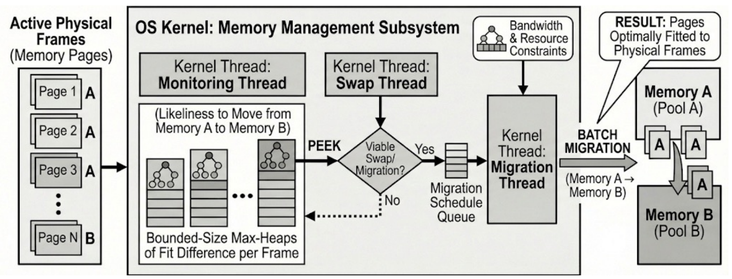


- Extends the Berkeley BOOMv3 RISC-V core with the SCTW accelerator as an MMIO device and last layer cache TileLink client.
- SCTW is a sidecar accelerator, non-blocking and off critical path, that writes memory fit updates directly to the last layer cache with eventual consistency.
- Maintains a ring buffer of score updates and generates a notification Interrupt for OS kernel handling.

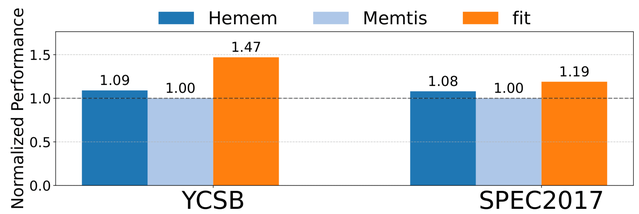
OS Implementation

We modify the Linux kernel to maximize alignment between page and memory. Key aspects of the design include:

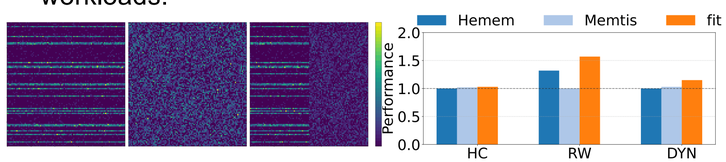
- OS Interrupt handling of fit scores provided by hardware.
- Maintains bounded-size heaps to keep O(logn) runtime per operation of the swap operator that migrates a page or swap two pages for better mapping
- Computes a locally optimal solution that is at least ½ of the global optimum, with migration rate dynamically tuned by bandwidth and resource availability.



Results



- In trace-level simulations with configured Optane and DDR5, our technique outperforms existing techniques from 11% (SPEC2017 Mixture) to 38% (YCSB Mixture).
- We attribute this gain to better handling of asymmetric read/write workloads, while maintaining parity in hot-cold workloads.



Federation of Experts: Communication Efficient Distributed Inference for Large Language Models

Muhammad Shahir Abdurrahman
Chun Deng, Philip Levis, Azalia Mirhoseini

Introduction

- Large Language Models (LLMs) consistently improve as parameter counts increase, but computation scales proportionally, limiting both training and inference. Mixture-of-Experts (MoE) architectures mitigate this by sparsely activating only a subset of experts per token, decoupling total parameters from active parameters.
- These efficiency gains, however, introduce a new bottleneck: the sparse routing of expert parallelism. When a token is assigned to an expert on a different GPU, it must be dispatched across the network, accounting for 68%+ of end-to-end latency. In DeepSeek's V3/R1 production deployments, for instance, 256 experts are distributed across up to 18 nodes to achieve higher throughput. This worsens in multi-node settings, where intra-node NVLink (900 GB/s) far outpaces inter-node InfiniBand (50 GB/s), a gap of roughly 18x.
- We introduce the **Federation of Experts (FoE)** architecture to reduce this bottleneck. FoE restructures attention and MoE FFN layers into H independent groups, with per-group routing that keeps the expensive all-to-all traffic local to each group.

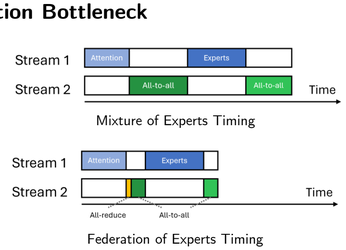
This paper makes three contributions:

1. We describe the FoE architecture, which partitions experts and KV heads into groups to reduce communication and guarantee even load across groups.
2. We show how FoE obtains an order-of-magnitude reduction in communication for both single-node and multi-node inference.
3. We demonstrate that on the LongBench dataset, FoE has up to 5.2x lower forward-pass latency, 3.62x lower time-to-first-token (TTFT), and 1.95x lower time-between-tokens (TBT) than an equivalent MoE model, while matching its generation quality.

Design

Tackling the Communication Bottleneck

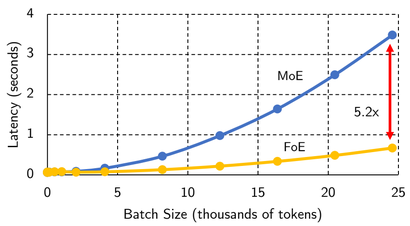
- Federation of Experts (FoE) reduces communication and improves load balancing by restructuring the MoE architecture.
- FoE additionally uses all-reduce which saves significant bandwidth over traditional all-to-all.
- Total forward-pass latency is significantly reduced.



Federation of Experts

- FoE restructures the attention and FFN layers into $H = n_{kv}$ independent MoE groups. For each group we have:
  - Independent attention layer $1/H$ of the KV heads
  - Expert group with $1/H$ of the total experts
  - Selects k/H experts exclusively within the expert group
- Results are synchronized between groups using a mean
  - Single-node: all-to-all is eliminated, only all-reduce remain
  - Multi-node: all-to-all intra-node, all-reduce inter-node
- Communication is reduced by an order of magnitude by co-locating memory and compute.

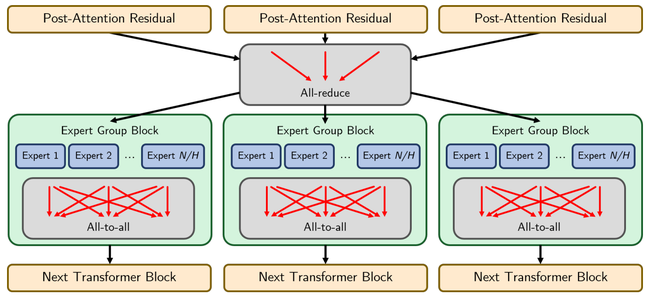
Forward Pass Latency Results



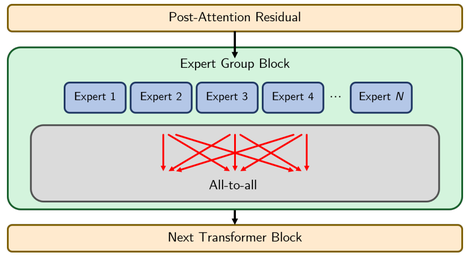
- We run synthetic forward passes at fixed batch sizes.
- FoE shows dramatically improved forward-pass latency as batch size increases.
- Demonstrates the effect of improved load balancing and reduced communication.

Architecture Overview

Federation of Experts



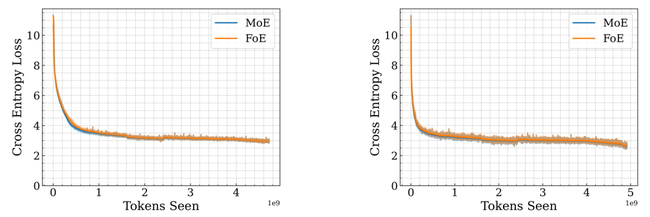
Mixture of Experts



Experimental Setup

- Hardware: 8x H100 nodes, 80GB VRAM per GPU, SXM + Infiniband.
- Training: We train MoE and FoE on our own training engine built from the ground up based on TorchTitan.
- Inference: We implement both MoE and FoE on our own inference engine FlexServe built from the ground up based on vLLM & SGLang. This properly isolates architectural gains from framework-level engineering disparities.
- Models: We train 1B & 7B parameter models for both FoE & MoE and report end-to-end inference results for the 7B model.
- Datasets: We evaluate on a variety of standard benchmarks for training and various LongBench tasks for inference.

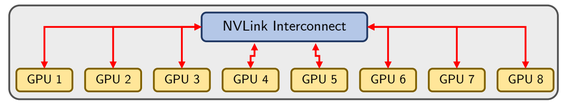
Training Results



Size	Variant	ARC-Easy	BoolQ	COPA	HellaSwag	PIQA	SciQ
1B	MoE	53.5%	57.7%	62.0%	30.9%	63.9%	77.7%
	FoE	52.7%	61.4%	61.0%	30.1%	63.8%	74.6%
7B	MoE	59.6%	60.4%	65.0%	33.7%	66.9%	80.0%
	FoE	58.2%	60.9%	66.0%	33.1%	66.6%	80.0%

Inference Results

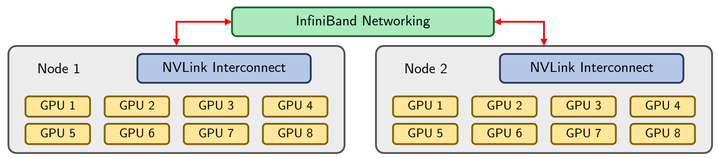
Single Node Setup



Single Node Speedup Results

Poisson Scale	Metric	Mean	p50	p99
0.5	Time to First Token	*2.66x*	*2.99x*	*1.82x*
	Time Between Tokens	*1.38x*	*1.31x*	*1.53x*
	End to End Time	*1.39x*	*1.32x*	*1.54x*
0.75	Time to First Token	*3.11x*	*3.21x*	*2.77x*
	Time Between Tokens	*1.83x*	*1.90x*	*2.06x*
	End to End Time	*1.85x*	*1.91x*	*2.07x*
1.5	Time to First Token	*2.87x*	*2.85x*	*3.15x*
	Time Between Tokens	*1.11x*	*1.08x*	*1.53x*
	End to End Time	*1.13x*	*1.09x*	*1.56x*

Multi Node Setup



Multi Node Speedup Results

Poisson Scale	Architecture	Mean	p50	p99
0.5	Time to First Token	*3.44x*	*3.57x*	*2.74x*
	Time Between Tokens	*1.36x*	*1.33x*	*1.57x*
	End to End Time	*1.37x*	*1.34x*	*1.57x*
0.75	Time to First Token	*3.62x*	*3.59x*	*3.70x*
	Time Between Tokens	*1.95x*	*2.01x*	*2.26x*
	End to End Time	*1.96x*	*2.02x*	*2.27x*

Limitations

- Benefits only apply to distributed deployments
  - On a single GPU, there is no all-to-all communication to reduce, thus the cross-group sums and separate attention layers become additional overhead without offsetting savings.
- Training speed is not improved
  - Each expert group has its own residual stream, resulting in a significant backward-pass overhead

Conclusions

- FoE achieves a breakthrough in MoE inference performance by restructuring the architecture to reduce communication and improve load-balancing.
  - FoE divides experts, attention and routing into groups thus enabling compute and memory to be co-located.
- Communication does not always equal model quality.
  - Existing architectures should undergo ablation experiments to understand whether any expensive data movement is truly needed for good generation.
- Next Steps:
  - Validate model quality scaling to larger parameter counts.
  - Can an existing trained MoE model checkpoint be adapted to an FoE model?