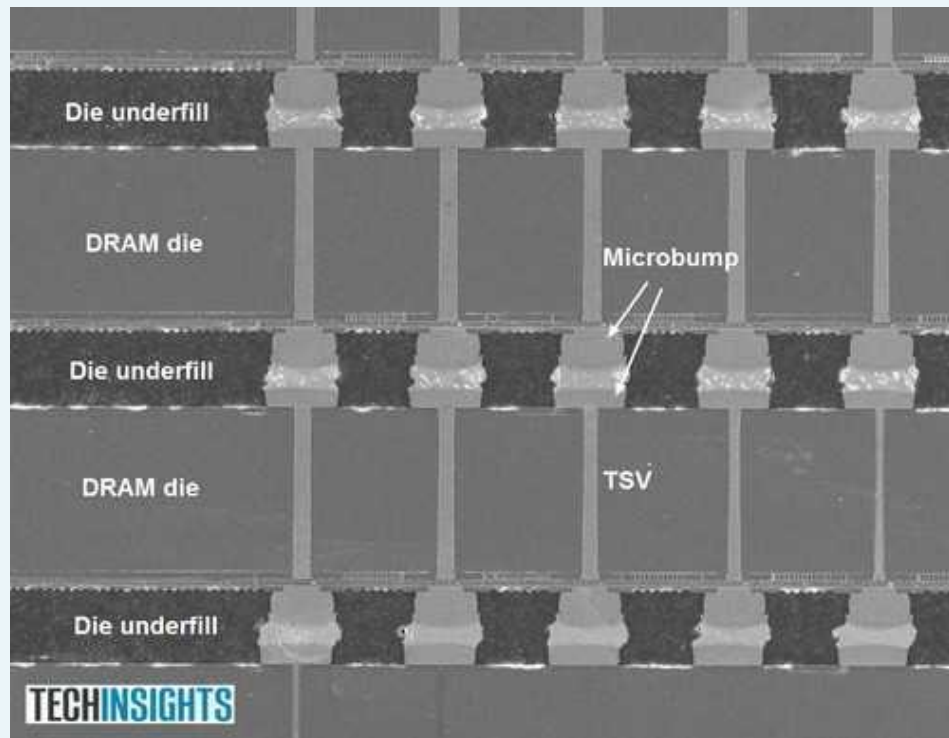
THE UNIVERSITY of EDINBURGH
informatics

Memory-Stacked GPU Architecture for LLM Inference

Zeyu Xu, Boris Grot

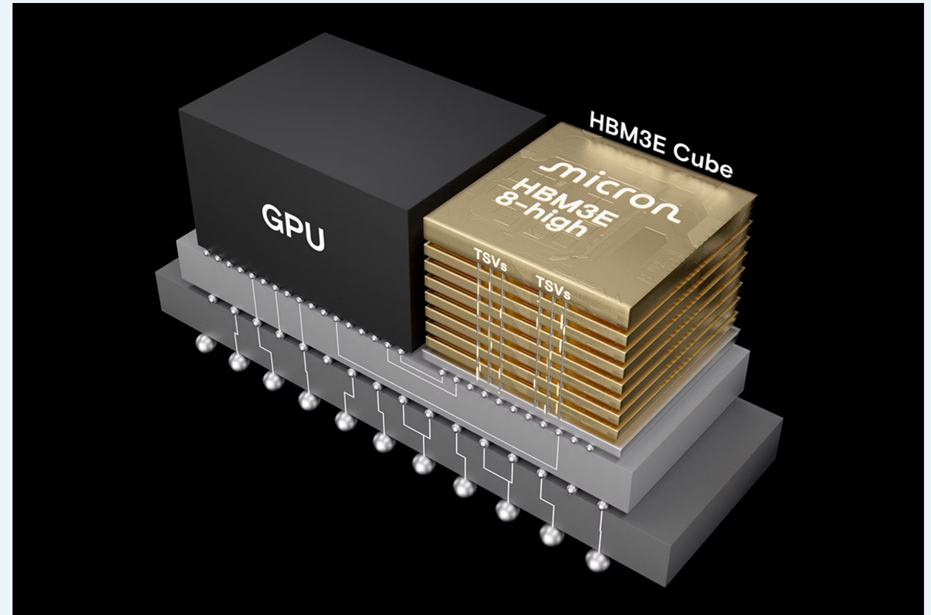
3D Die Stacking

- Through Silicon Via (TSV) drills through logic dies
- Microbumps form contact connecting metal layers on different dies



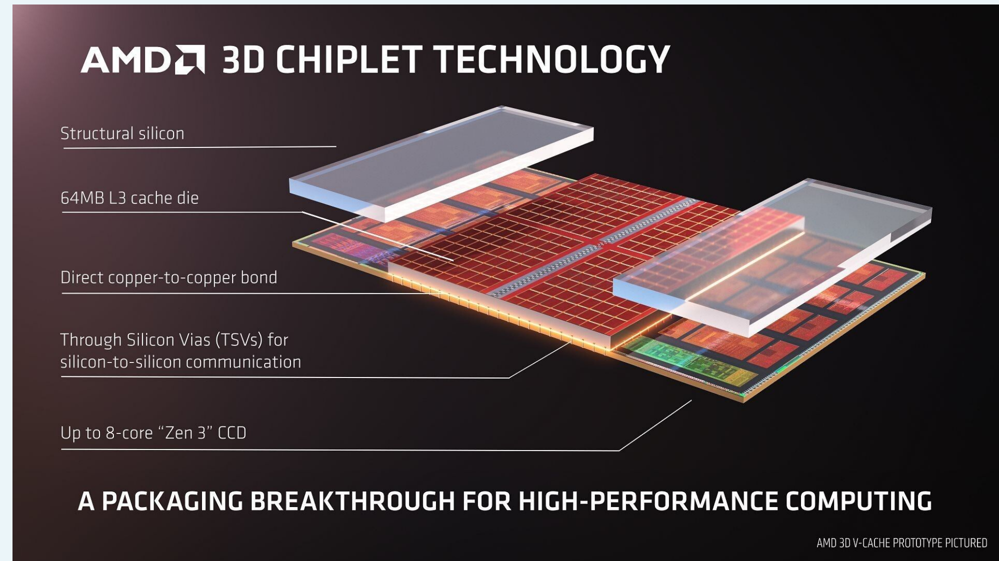
3D-Stacked Commercial Products

- HBM
  - Stacked DRAM layers
  - Energy-efficient, high bandwidth memory



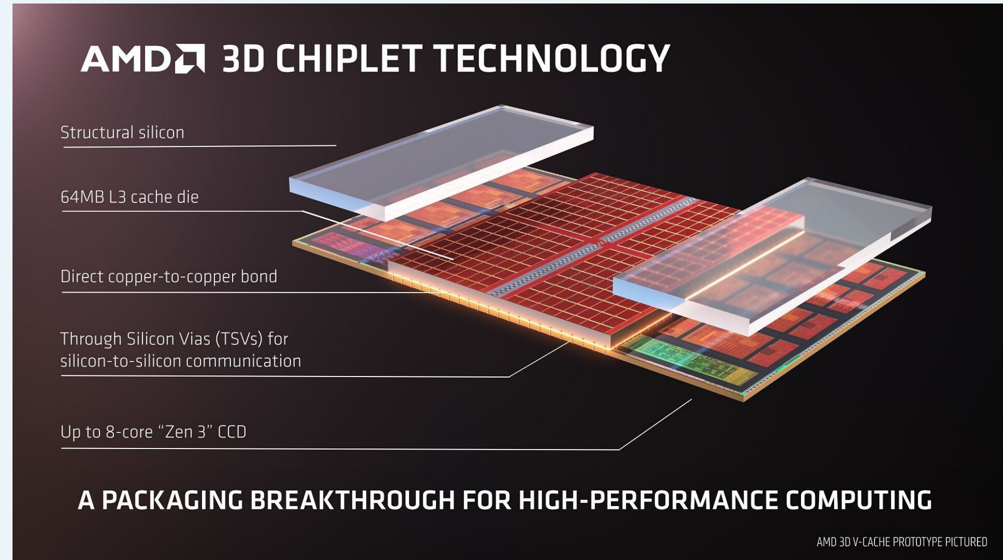
3D-Stacked Commercial Products

- 3DV Cache
  - L3 Cache die on top of CCD



3D-Stacked Commercial Products

- 3DV Cache
    - L3 Cache die on top of CCD



Our Vision: Further demonstrate the potential of 3D stacking through more drastic architectural changes

Memory-Stacked GPU Architecture

- TSVs have low connection density
  - Cannot split fine-grain structures across dies
  - Memory stacking

Connection	TSV	2D wire
Pitch	20um[1]	50nm[2]

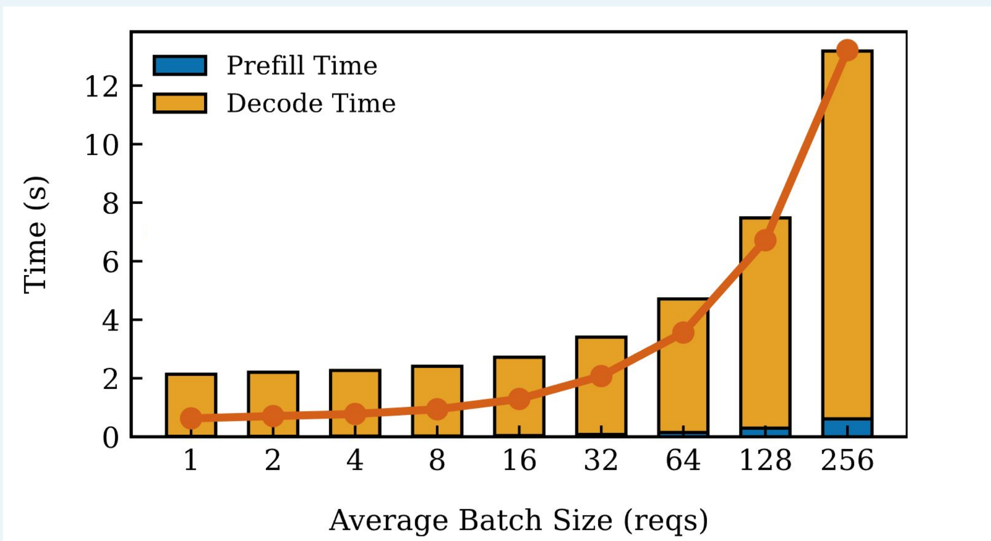
1. K. -I. Moon, et. al., IEDM 23
2. S. Liao, et al., IEDM 24

Memory-Stacked GPU Architecture

- GPUs are relatively simple
  - Replicated structures
  - Concentrated workloads

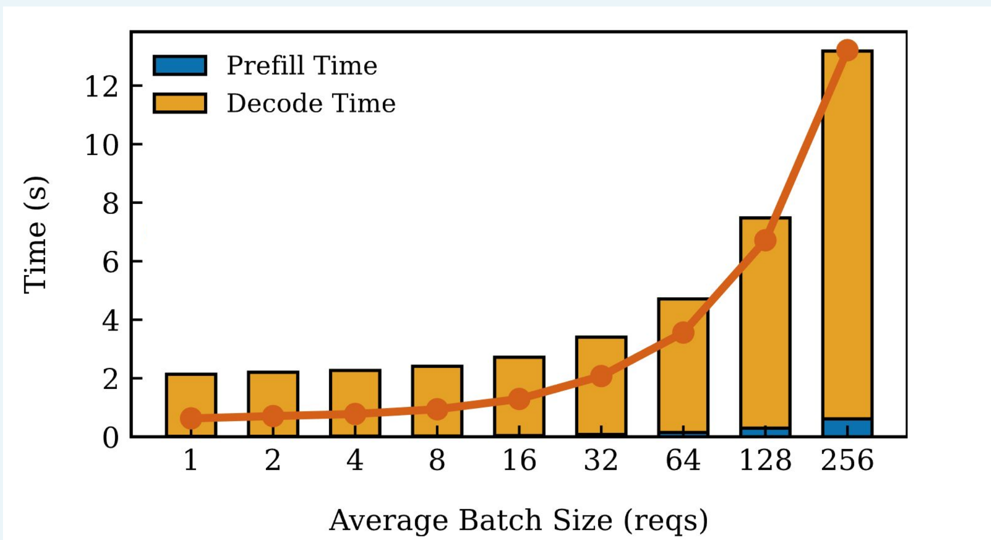
Workload: LLM Inference

- Prefill Phase
    - Process input prompt
    - Compute intensive
    - Small fraction of ex time



Workload: LLM Inference

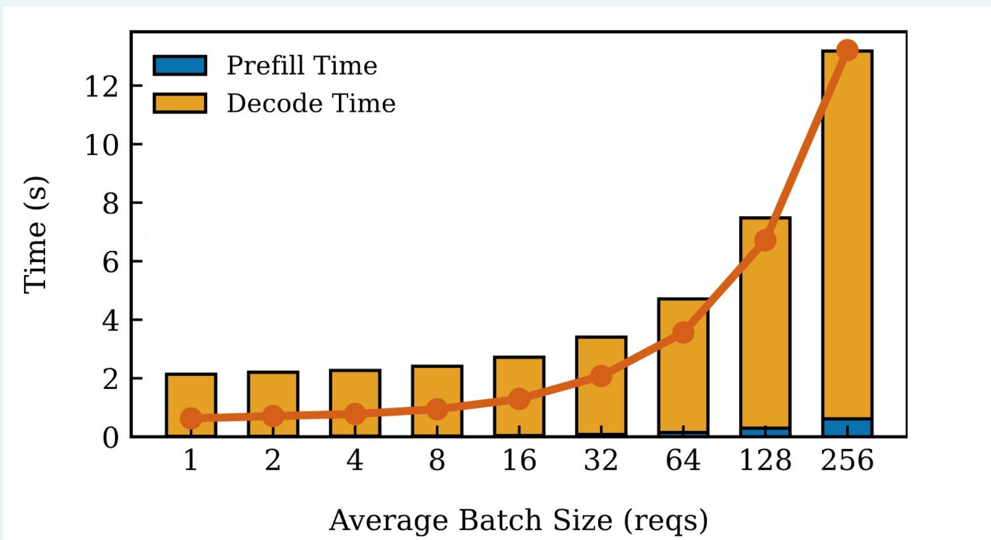
- Decode Phase
  - Output generation
  - Memory bound
  - Majority of ex time



THE UNIVERSITY of EDINBURGH
informatics

Workload: LLM Inference

- Decode Phase
  - Output generation
  - Memory bound
  - Majority of ex time



Disaggregated Serving: Use separate devices for prefill and decode

Memory-Stacked GPU for LLM Decode

- Stack layer as a NUMA node
  - Lower latency
  - Higher capacity
  - Higher bandwidth

Memory Latency Reduction

- GPU mem latency ~300ns
- Stack layer can reduce ~200ps latency[1]
- GPU architecture is latency tolerant

Delay (ns)	0	360	1800
Ex Time (us)	73.9	77.9	268.9
Diff (%)	-	5.4	263.9

1. H. Suh, et. al., EPEPS 24

Memory Capacity Increase

- With larger capacity
  - Serve larger, more capable models
  - Longer context
  - Batch more requests at each step

THE UNIVERSITY of EDINBURGH
informatics

Memory Capacity Increase

- With larger capacity
  - Serve larger, more capable models
  - Longer context
  - Batch more requests at each step

- To increase capacity
  - Multiple GPUs per node
  - Multiple nodes per model

  Nx increase

Memory Capacity Increase

- Additional memory from stacking
  - ~1GB of SRAM
  - ~20GB of DRAM
- H100 has 80GB of HBM
  - 25% increase
  - Not cost-effective
- To increase capacity
  - Multiple GPUs per node
  - Multiple nodes per model

Nx increase

Memory Bandwidth Increase

- Decode has low arithmetic intensity
  - Optimal value for H100 is ~141
  - Behaves like data scan

2x bandwidth => 2x speedup

Layer Name	OPs	Memory Access	Arithmetic Intensity	Max Performance	Bound
Decode					
q_proj	34M	34M	1	768G	memory
k_proj	34M	34M	1	768G	memory
v_proj	34M	34M	1	768G	memory
o_proj	34M	34M	1	768G	memory
gate_proj	90M	90M	1	768G	memory
up_proj	90M	90M	1	768G	memory
down_proj	90M	90M	1	768G	memory
qk_matmul	17M	17M	0.99	762G	memory
sv_matmul	17M	17M	0.99	762G	memory
softmax	328K	262K	1.25	960G	memory
norm	29K	16K	1.75	1T	memory
add	4K	16K	0.25	192G	memory

Memory Bandwidth Increase

- Decode has low arithmetic intensity
    - Optimal value for H100 is ~141
    - Behaves like data scan

2x bandwidth => 2x speedup

- 132 SMs in H100
- Memory can saturate ~ 1 SM

Memory Bandwidth Increase

- Decode has low arithmetic intensity
  - Optimal value for H100 is ~141
  - Behaves like data scan

2x bandwidth => 2x speedup

- 132 SMs in H100
- Memory can saturate ~ 1 SM

Replace surplus compute with TSVs for high memory bandwidth?