A Fast Locality Simulator for GEMM Design-Space Exploration on Multi-Chiplet GPUs

Abstract

Euijun Chung, Hyesoon Kim

HPArch @ School of Computer Science, Georgia Institute of Technology

HPArch @ Georgia Tech Georgia Tech

1. Background

Multi-chiplet GPUs have multiple NUMA regions.

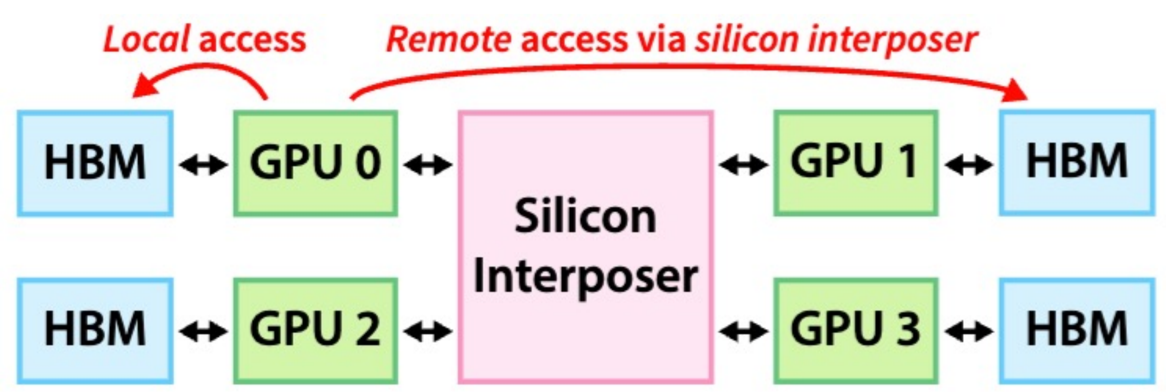
- E.g., AMD MI300X has **four I/O dies** (IODs)
- *Remote* HBM accesses cross the **silicon interposer**

However, *remote* HBM accesses incur:

① High **energy consumption** (local 3.5 → remote 4.9 pJ/bit)

② Longer **access latency**

③ **Inter-chiplet bandwidth** consumption



Memory locality is critical for performance and energy efficiency on **multi-chiplet GPUs**.

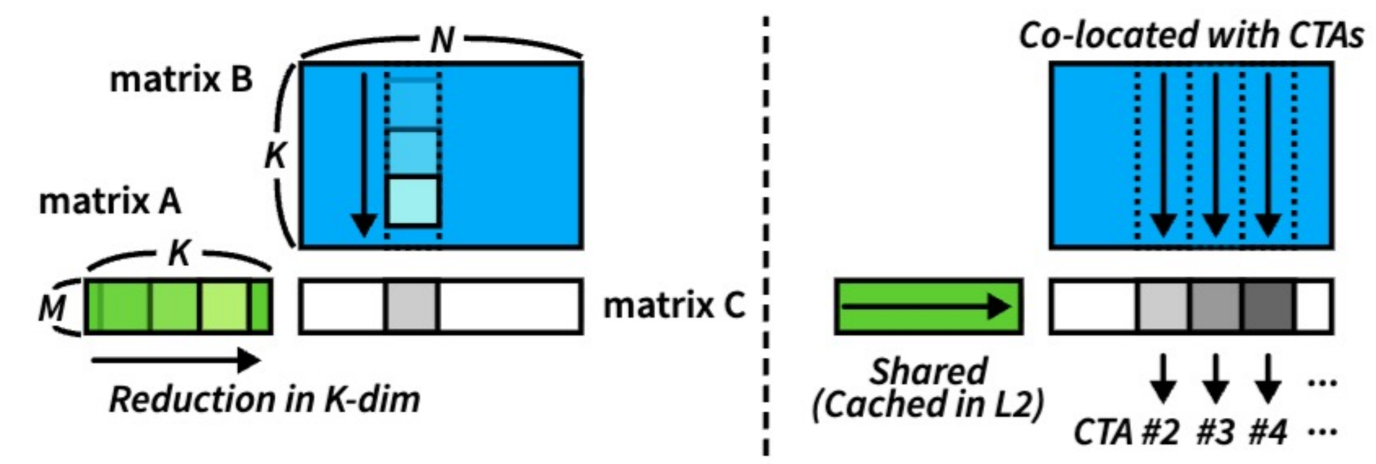
2. Motivation

Locality-aware data & CTA placement co-locates data and compute on the same chiplet to minimize remote HBM traffic.

- Requires changing `malloc()` and CTA scheduling HW logic

GEMM (General Matrix Multiply) is a natural target for locality-aware placement due to its tile-level access pattern.

- Dominant operation in LLM inference & training



(a) (M, N, K) Matrix-Multiplication (b) CTA-aware data placement

GEMM is **dominant in LLMs**, and its **regular access pattern** makes locality easy to characterize.

3. Challenges

Best placement strategy on Llama 70B reduces remote HBM traffic by **18.3×** vs. 4KB interleaving and **58×** vs. worst case.

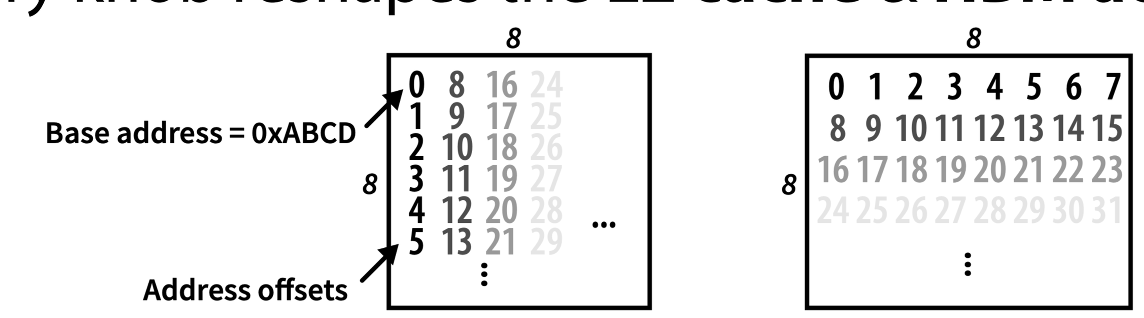
However, due to the **large design space**, finding the best placement is **non-trivial**:

① GEMM **dimensions** (M, N, K) & **tile sizes** (T_M, T_N, T_K)

② **Data layout** (row- / column-major) per matrix

③ **CTA traversal order** (row / column / 2D block-swizzle)

→ Every knob reshapes the **L2 cache** & **HBM access pattern**.



(a) Column-major layout (b) Row-major layout

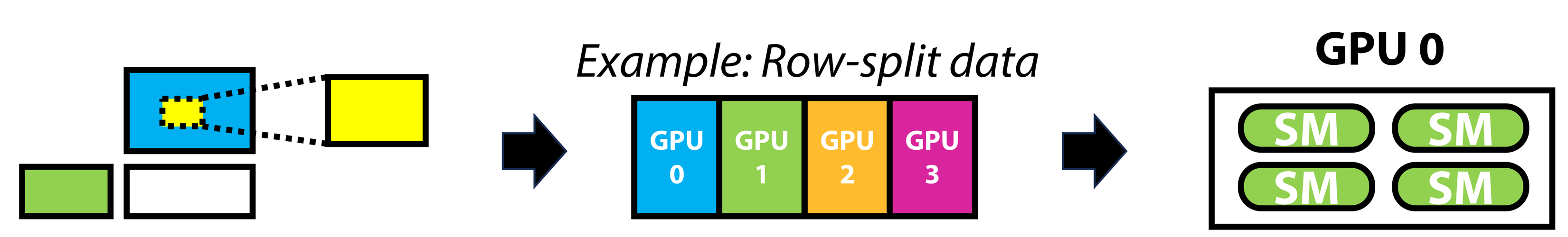
Finding the **best locality-aware placement** for GEMMs is **important**, but **hard**.

4. Simulator design

Tile-granularity GEMM **execution** and **memory accesses**.

- Per-chiplet **L2 cache** slices
- **Performance** & **power** with roofline analysis

→ *Fast simulation* with no cycle-level detail required.



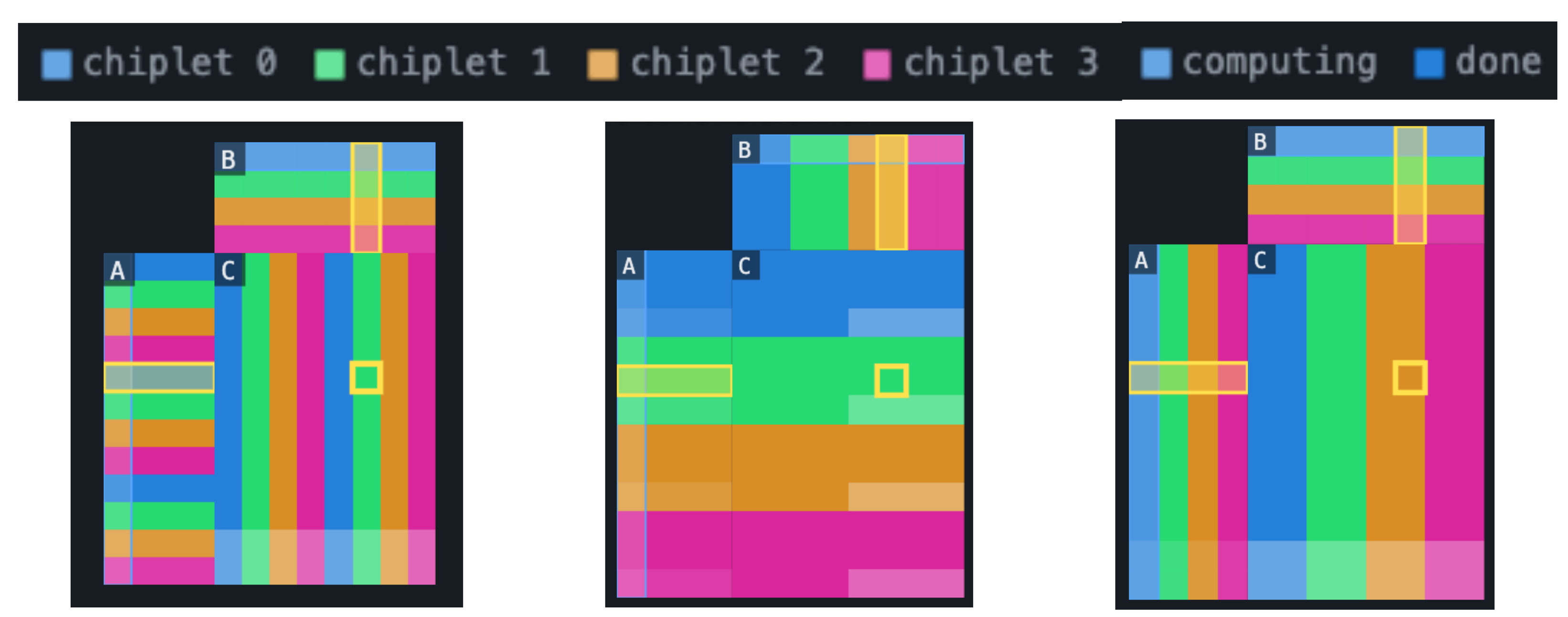
1. **GEMM kernel**
   - Dimensions
   - Tile sizes
   - CTA traversal
2. **Data placement**
   - 4KB interleaving
   - Row-split
   - Column-split
3. **CTA execution**
   - CTA-to-chiplet mapping
   - SRAM / L2 cache

Output: L2 hit rate, local/remote HBM traffic (GB), perf, power.

Tile-granularity GEMM simulation enables fast exploration of placement strategies.

5. Locality-aware GEMM

The simulator can **evaluate** and **visualize** multiple data & CTA placement strategies on **various GEMM configurations**.



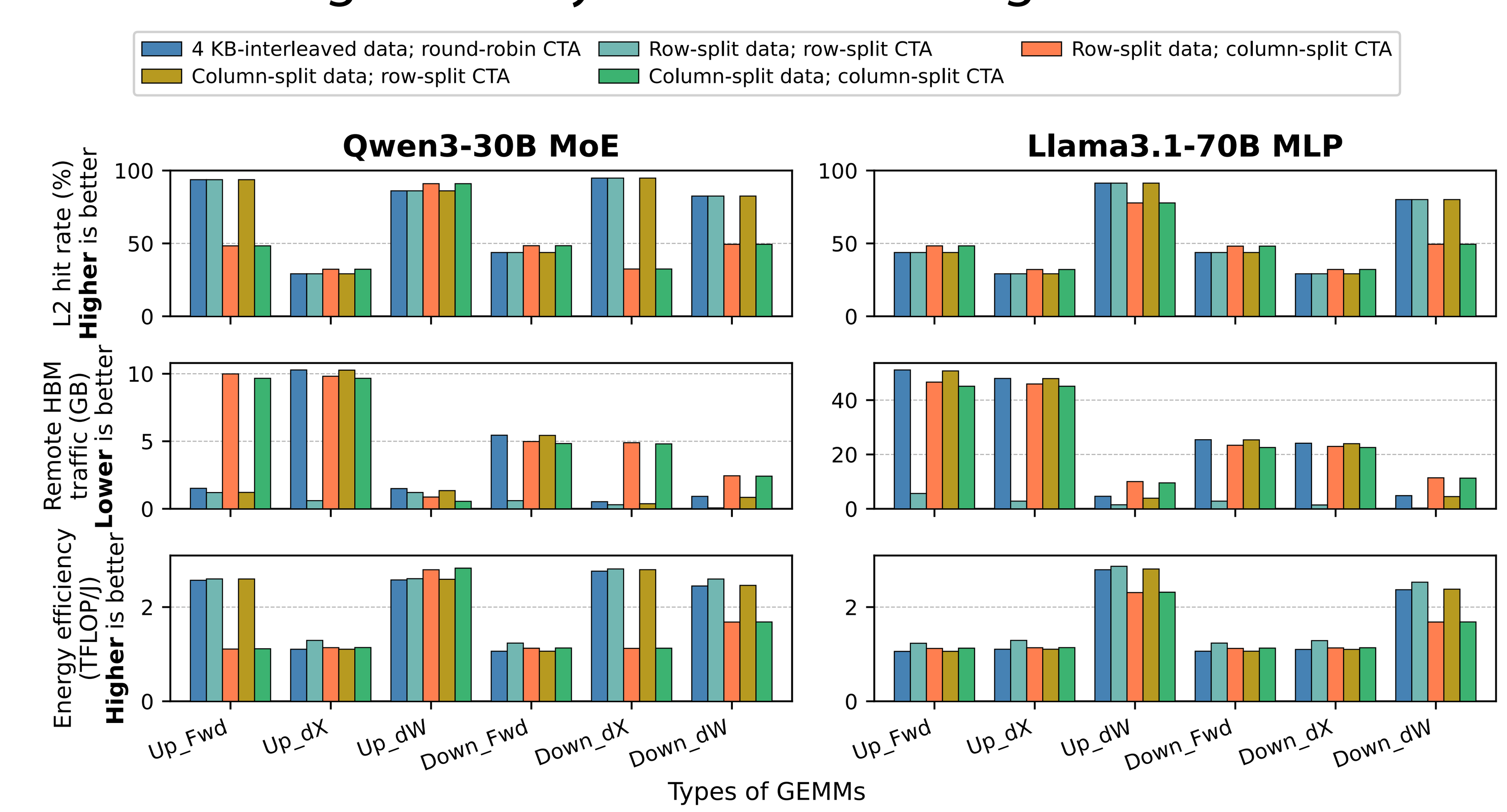
4KB-granularity data interleaving | Optimal data & CTA placement | Sub-optimal data & CTA placement

Future work: extension to attention and convolution kernels.

The simulator provides insights into **L2 cache** & **HBM access patterns** based on **locality-aware placements**.

6. Evaluation

Locality-aware placement reduces **remote HBM traffic** by up to **18.3×** and improves **perf-per-power** by up to **17%** over to 4KB-granularity data interleaving.



Our simulator can evaluate **locality, performance, and power** of data & CTA placement strategies.