# Lecture 4 : Announcements

- hw time spent
- office hours
- don't forget: talk topic + SSH key
- talk time assignment

# Case Study: Streaming Workloads

Q: Estimate epxected throughput for `saxpy` on an architecture with caches. What are the right units?

$$z_i = \alpha x_i + y_i \quad (i = 1, \ldots, n)$$



Demo: https://github.com/lcw/stream_ispc

# Special Store Instructions

"Non-temporal stores"

At least two aspects to keep apart:

- Non-temporal locality
- Spatial locality

What hardware behavior might result from these aspects?

- Non-temporal: writing past cache, hint at fastest
- Spatial: don't bring in data, mark rest of line undef

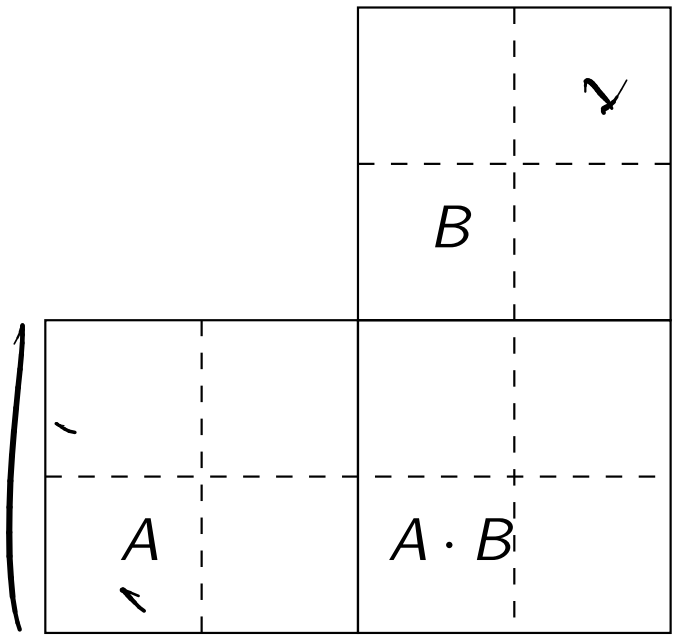
- Comment on what a compiler can promise on these aspects.
- Might these 'flags' apply to loads/prefetches?

(see also: [McCalpin '18])

# Case study: Matrix-Matrix Mult. ('MMM'): Code Structure

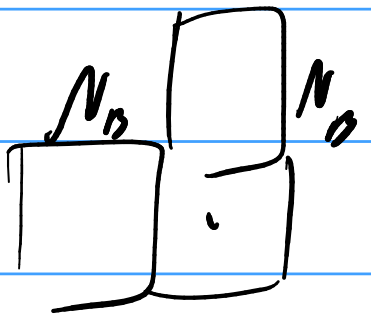
$O(n^2) - O(n^3)$

- How would you structure a high-performance MMM?
- What are sources of concurrency?
- What should you consider your working set?




concurrency: absence of ord. req. | parallelism: ↓↓

$(a+b)+c+(d+e)$



# Case study: Matrix-Matrix Mult. ('MMM') via Latency

Come up with a simple cost model for MMM in a two-level hierarchy based on latency:

$N$ $N_B$

Avg latency per access =

(1 − miss ratio) · cache latency + miss ratio · mem lat.

Total accesses: $4N_B^3$

Misses: $3N_B^2$

Miss ratio: $\frac{3N_B^2}{4N_B^3 \cdot \text{cache line size}} = \frac{3}{4N_B \cdot CLS}$

[Yotov et al. '07]

# Case study: Matrix-Matrix Mult. ('MMM') via Bandwidth

- Come up with a cost model for MMM in a two-level hierarchy based on bandwidth:

16 × 2 FMAs

512/32 = 2 FMA-capable exec units

Fused Multiply Add

Cycle count for whole calc: $2N^3/(2 \cdot 32)$

$4N^3/(N^3/32) = 128$ floats/cycle

Total mem ⇔ cache data motion:

# blocks · (block size) · 4 = $(N/N_B)^3 \cdot N_B^2 \cdot 4$

Required mem bw:

$4N^3/N_B/(N^3/32) = 128/N_B$ floats/cycle

total mem. motion / #cycles

[Yotov et al. '07]

# Case study: Matrix-Matrix Mult. ('MMM'): Discussion

Discussion: What are the main simplifications in each model?

[Yotov et al. '07]

General Q: How can we analyze cache cost of algorithms in general?