February 11, 2025

Announcements

- hw1
  - talk assignment posted

Goals

- Cache mysteries
- Streaming workloads
- gemm: modeling
- caches + program structure

Review

- Cache features
  - lines
  - associativity
  - hierarchy

Flops are cheap

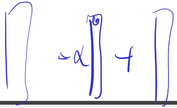
Bandwidth is money

Latency is physics

Case Study: Streaming Workloads

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

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



Units: GBytes/s

Net memory accessed: $3 \cdot 4 \cdot n$

$4 \cdot 4 \cdot n$

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

Special Store Instructions

At least two aspects to keep apart:

– temporal locality: cache, don't keep

– spatial locality: entire line will be overwritten

What hardware behavior might result from these aspects?

– invalidating cache

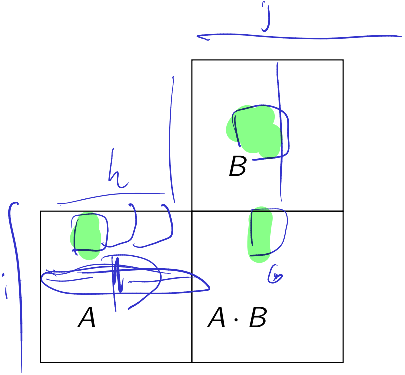
– not fetching the cache line

- 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

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



Sources of concurrency: i, j, k?

Working set sizes:

matched to memory hierarchy

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

Cost model for MMM in a two-level hierarchy based on latency?

read A ✓

-"- B ✓

-"- C ✓

write C ✗

miss

Total accesses: $4N_B^3$

Misses: $3N_B^2$

Miss rate: $\frac{\#\text{Misses}}{\#\text{accesses}} = \frac{3}{4N_B}$

per tile

[Yotov et al. '07]

```
for i
    for j
        for k
            C[i,j] += A[i,k] * B[k,j]
```

Avg. latency per access =

$= (1 - \text{miss rate}) \cdot \underline{\text{cache latency}}$

$(\text{miss rate}) \cdot \underline{\text{DRAM latency}}$

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

Cost model for MMM in a two-level hierarchy based on bandwidth?

[Yotov et al. '07]