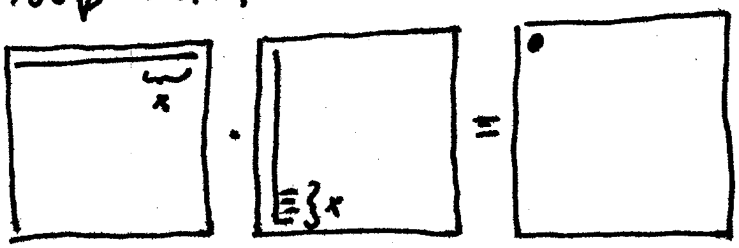
Why blocking?

assume: cache size $\ll n$
cache line = 8 doubles
only 1 cache

CM = cache miss

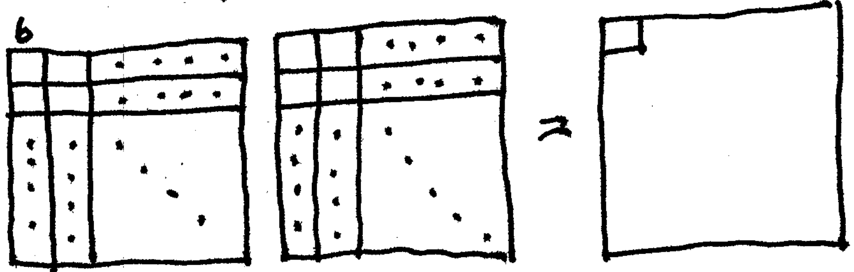
1.) Triple loop MMM



1. entry: $\frac{n}{8} + n$ CMs (compulsory)
afterwards: x is in cache
2. entry: no reuse, so again $\frac{n}{8} + n$ CMs

$$\Rightarrow \text{total} = \left(\frac{n}{8} + n\right) n^2 = \frac{9}{8} n^3 \text{ CMs}$$

2.) blocked MMM



choose: $b \geq 8$ (cacheline)
and $8 \mid b$
and $3b^2 \leq c$
c = cache size

1. block: $\frac{nb}{8} + \frac{nb}{8} = \frac{nb}{4}$ CMs

2. block: same

$$\Rightarrow \text{total} = \frac{nb}{4} \cdot \left(\frac{n}{b}\right)^2 = \frac{n^3}{4b}$$

choose $b = \sqrt{\frac{c}{3}}$ $\Rightarrow \frac{\sqrt{3}}{4\sqrt{c}} n^3$ CMs

gain: $\approx 2.5\sqrt{c}$

- Explains much of triple loop's poor performance (the other major optimization is unrolling and scalar replacement for better instruction parallelism and register usage)
- Blocking achieves both: better spatial and better temporal locality with respect to the cache
- In 2.) the number of cache misses = amount of data transferred cache <-> memory is O(n^3/sqrt(C)). Hence the operational intensity is O(sqrt(c)). It is known that this is optimal, i.e., Theta(sqrt(c)).