Aug 31st

Physical locality ← transfer cost
Temporal locality ← re-use
Correlation locality

Metrics:

hit rate

composite performance metrics – access time

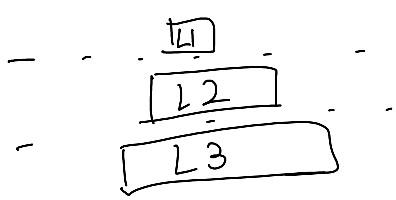
arithmetic intensity = $\frac{\#\text{ops}}{\text{data transferred}}$

$$\boxed{\text{"Locality"} = \frac{\#\text{ words accessed locally}}{\#\text{ words accessed for computation}}}$$

$$= \frac{\text{total} - \text{transfers}}{\#\text{ ops} \times \text{operands/operation}} \qquad \left(\frac{\#\text{ bytes}}{\#\text{ bytes}}\right)$$

⇒ #ops ⇒ ISA dependent ⇒ Keep it at 'C' level like ADD, MUL

What does "locally" mean?



$P \leftrightarrow M$

⇒ Put a line in hierarchy; above that is local.

⇒ Depends on the point of view.

Correlation locality

→ Hit rate may include correlation locality (Paper — Ask Mattan)

EXAMPLE: MAT-MUL

$N\,\boxed{A}_N \times \boxed{B} = \boxed{C}$

Look at the hit rates.

Counting misses w.r.t. a cache.

for i
  for j
    for k

$C_{ij} += A_{ik} B_{kj}$

First Time: $(2k+1)$ locations needed to store entire row of A (k), Entire column of B (k) & C_{ij} (1).

Now move to next column of B.

Cache size Z

Parameters: Fully Associative
Cache size = Z
Line Size = 1
Replacement Policy = LRU
Write-back
Allocation: R/W (anything we touch, put into cache)
Prefetcher: None.

$\Rightarrow$ $Z > 2N+1$ (Keep entire row of A) $\Rightarrow$ How can we get ~~N+1~~ with different replacement policy ??

Any ideal replacement policy is within 2x of LRU.

Total no. of Misses: $N \cdot N \cdot (2N+1) = 2N^3 + N^2$ $\leftarrow$ transferred.

total $= N \cdot N \cdot N \cdot (4) \longrightarrow$ (old C, new C, A, B)

$$\frac{4N^3 - (2N^3 + N^2)}{4N^3} \to \frac{1}{2}$$ Not very good.

If we have Cache Line Size = L

$$\text{no. of misses} = \frac{N+1}{L} + N$$

$$\#\text{ of transfers} = \left(\frac{N^3}{1} + \frac{N^3}{L} + \frac{N^2}{L}\right) \cdot L$$

Things get worse because 'L' comes into picture, since B has no cache line locality.

Change iteration order

For i
For j $\longrightarrow$ (can interchange

For i
For k

For k → optimization

for k

for j

$C_{ij} = A_{ik} * B_{kj}$ $\qquad$ $C_{ij} += A_{ik} * B_{kj}$

→ Now we have more misses on C

⇒ Compilers might do this optimization automatically.
(loop inter-change)

-no fast_math

<u>if $Z > 2N+1$:</u>

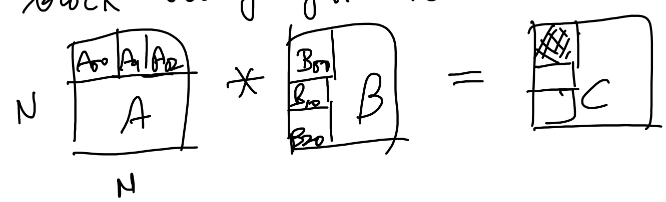
now instead of $2N^3 + N^2 = N^2(2N+1)$

we get better locality. locality $= N^2(N+2)$

if <u>$Z > 3N^2$:</u> all matrices (Cache fits all matrices)

locality $= \frac{-3N^2 + 4N^3}{4N^3}$ → going towards 1

Take a block out of full matrices:



$C_{00} = A_{00}B_{00} + A_{01}B_{10} + A_{02}B_{20}$

↓

<u>b×b sub matrix</u>

Did it improve locality?

Choose b such that $Z > 3b^2$

$b < \sqrt{\frac{Z}{3}}$

$b^2 + 2b^2 + 2b^2 + 2b^2$

b^2 ↓ Bring C; $2b^2$ ↓ Bring $A_{00}\,B_{00}$; $2b^2$ ↓ $A_{01}\,B_{10}$

$= b^2 + \sum_{0}^{N/b} 2b^2 = b^2 + \frac{2N}{b} \cdot b^2$

$$= (b^2 + 2Nb) \cdot \left(\frac{N}{b}\right)^2 = N^2 + \frac{2N^3}{b} = N^2 + \frac{2\sqrt{3}N^3}{\sqrt{Z}}$$

⇓ Total no. of subblocks

Earlier: $2N^3 + N^2$

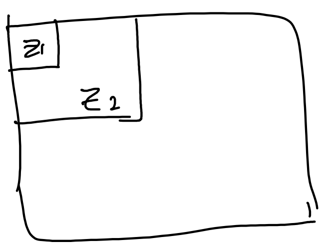
Now: $\frac{2\sqrt{3}N^3}{\sqrt{Z}} + N^2$ → improved by $\sqrt{Z}$ times

Now Locality $= \dfrac{4N^3 - \left(\frac{3.4N^3}{\sqrt{Z}} + N^2\right)}{N^3}$

Hit Rate improves significantly by re-ordering.

→ This Technique is called Blocking (Tiling).

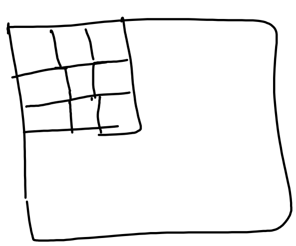
One more Cache: Z_1 Z_2



To gain Locality in L2, we'll incur misses in L1 but less ~~memory~~ load from memory.

→ optimize for L1 or L2 ??

Within L2 Block, optimize for L1.



Nesting Tiling

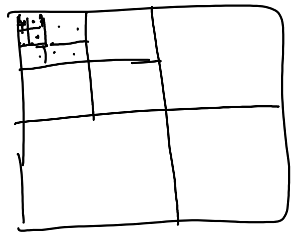
```
for ii
  for jj
      C_ii,jj = 0
      for kk
          for i = b.ii
              for j = b.jj
```

for k

How to come up with an order-oblivious of Z.

Base → 2x2 / 4x4

Start from top:



This is divide & conq algo.

$C_{00} = A_{00} \cdot B_{00} + A_{01} \cdot B_{10}$

$C_{01} = A_{00} \cdot B_{01} + A_{01} \cdot B_{11}$

$C_{10} = A_{10} \cdot B_{00} + A_{11} \cdot B_{10}$

$C_{11} = A_{10} \cdot B_{01} + A_{11} \cdot B_{11}$

→ Blend between Cache-aware & Cache-obliviousness;

→ stop recursion when subblock > thresh.

r – recursion depth

C – read $2^{r \sim \log(N)}$ times

no. of misses

$$Q(N) < 8 \cdot Q\left(\frac{N}{2}\right) \Rightarrow \Omega\left(\frac{N^3}{Z} + N^2\right)$$

⇒ Labs: Prefetcher ON → will change the performance

PIN 2

Performance Counters.