Model-based ATLAS (Yotov et al.)

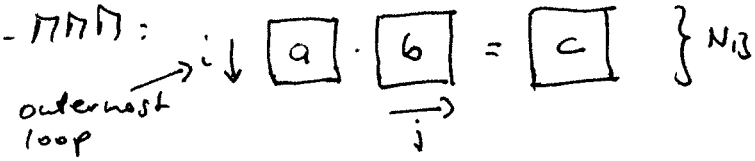
Basic idea: Use more hardware parameters than ATLAS and use models to predict mini-MMM parameters.

Additional hardware parameters: C_1 = L1 cache size
B_1 = cache block size
L_x = fl. point latency

Note: I work with ijk loop order, the paper with ijk (reason: in class we work with C, ATLAS with Fortran

1.) Estimate N_B
Assume cache is fully associative, refine model in steps
Idea: working set has to fit in cache

mini-MMM:



a.) easy bound: working set $= 3N_B^2$

$\Rightarrow 3N_B^2 \leq C_1$

b.) closer analysis

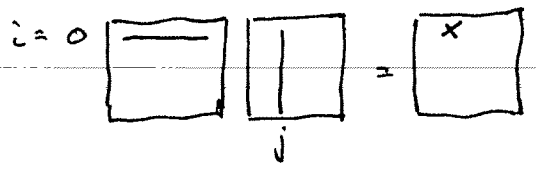
$N_B^2 + N_B + 1 \leq C_1$

↑ b ↑ row of a ↖ element of c

c.) take into account cache block size, i.e., units are cache blocks now

$$\left\lceil \frac{N_B^2}{B_1} \right\rceil + \left\lceil \frac{N_B}{B_1} \right\rceil + 1 \leq \frac{C_1}{B_1}$$

d.) take into account LRU replacement
build a history of array elements (say touched

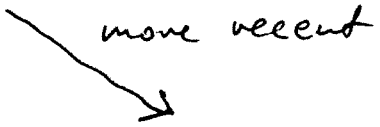


$i=0$:

$a_{00}\, b_{00} \ldots\ldots a_{0N_B-1}\, b_{N_B-1\,0}\quad c_{00}\qquad (j=0)$
$a_{00}\, b_{01} \ldots\ldots a_{0N_B-1}\, b_{N_B-1\,1}\quad c_{01}\qquad (j=1)$
$\ldots\ldots\ldots\qquad \ldots$
$a_{00}\, b_{0N_B-1} \ldots\ldots a_{0N_B-1}\, b_{N_B-1 N_B-1}\quad c_{0N_B-1}\qquad (j=N_B-1)$

corresponding history:

$b_{00} \ldots\ldots b_{N_B-1\,0}\quad c_{00}$
$b_{01} \ldots\ldots b_{N_B-1\,1}\quad c_{01}$
$\ldots\ldots\ldots\qquad \ldots$
$a_{00}\, b_{0N_B-1}\quad a_{0N_B-1} b_{N_B-1 N_B-1}\quad c_{0N_B-1}$



observations:
- all of b, starting with b_{00}, has to be in cache for $i=1$
- when $i=1$, row 1 of a will not clearly replace row 0 of a.
- when $i=1$, element of c will not clearly replace the previous elements of c.

$\Rightarrow$ This has to fit:
- entire b
- 2 rows of a (here: a_{0*}, a_{1*})
- 1 row of c (here: c_{0*})
- 1 element of c (here: c_{10})

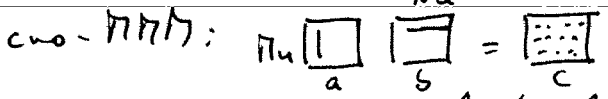
Result: $\left\lceil \frac{N_B^2}{B_1} \right\rceil + 3\left\lceil \frac{N_B}{B_1} \right\rceil + 1 \le \frac{C_1}{B_1}$

e.) take into account register tiling

$$\left\lceil \frac{N_B^2}{B_1} \right\rceil + 3\left\lceil \frac{N_B \pi_a}{B_1} \right\rceil + \left\lceil \frac{\pi_a \nu_a}{B_1} \right\rceil \le \frac{C_1}{B_1}$$

f.) shrink N_B so $\pi_a, \nu_a \mid N_B$ (avoids clean-up code)

2.) Estimate M_u, N_u

micro-MMM: M_u  =

a.) since scalar replacement loads entire column of a, ~~column~~ row of b and c is reused:

$$M_u N_u + M_u + N_u \leq N_R$$

b.) after shedding, L_s more registers are needed

$$M_u N_u + M_u + N_u + L_s \leq N_R$$

- solve assuming $M_u = N_u$, but both have to be at least 1
- adjust (increase) N_u if possible

So roughly $M_u \approx N_u$ (will perform poorly on Intel, more later)

3.) Estimate L_s

mul$_1$
⋮
mul$_{L_s}$
add$_1$
mul$_{L_s+1}$
add$_2$
⋮
add$_{L_s}$

2 L_s - 1 instructions in between

$$\Rightarrow L_s = \left\lceil \frac{L_x + 1}{2} \right\rceil$$

4.) Estimate K_u

Choose such that loop body fits into I-cache

Make sure $K_u \mid N_B$ (avoid clean-up code)

In most cases: $K_u = N_B$.