Model-Based ATLAS (Yotov et al.)

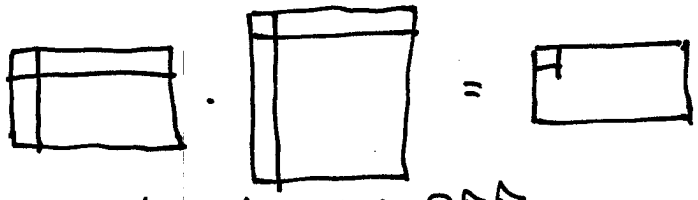
Uses microarchitecture parameters:

cache size C_1 } (L1 cache)
cache line size B_1 }
float mult latency L_x
.....

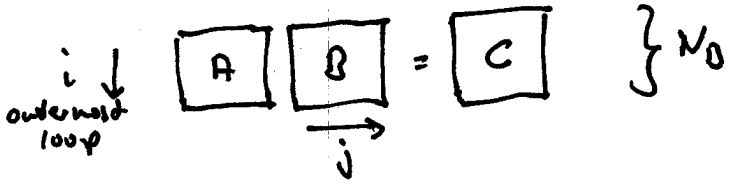
size in doubles

1.) Estimating N_B, assume cache is fully associative

MMM: N_B ε



blocked into mini-MMMs:



refine model step by step:

a.) working set is $3N_B^2$

$\Rightarrow 3N_B^2 \leq C_1$

b.) closer analysis

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

↑ whole B ↑ row of A ↖ element of C

c.) take cache line size into account
means: units are cache lines

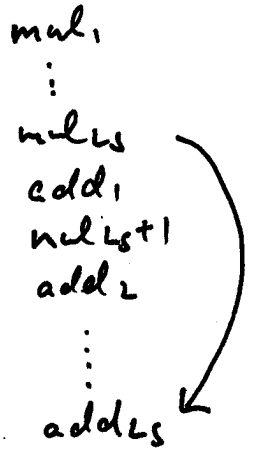
$$\left\lceil \frac{N_B^2}{B_1} \right\rceil + \left\lceil \frac{N_B}{B_1} \right\rceil + 1 \leq \frac{C_1}{B_1}$$

↑ cache lines for B

3.) Estimating K_u

- choose such that unrolled k'', i'', j'' loops fit into I-cache
- $K_u \mid N_3$ (avoids clean up code)

4.) Estimating L_s, assume float units are fully pipelined
means: can finish one operation per cycle



$2L_s - 1$ instructions in between
these should hide float mult latency L_x
(e.g. $L_x = 5$ on Core)

a.) $\Rightarrow$ $2L_s - 1 \geq L_x$ or $L_s \geq \left\lceil \frac{L_x + 1}{2} \right\rceil$

b.) n_x float mult units may be available

$\Rightarrow$ $\frac{2L_s - 1}{n_x} \geq L_x$ or $L_s \geq \left\lceil \frac{L_x n_x + 1}{2} \right\rceil$