Jacob Barzee

P2: Localize CNN

Version	Notes & File	Case	Seq Time	Par Time	Speed Up
Seq	No changes	1	5.778518	5.907502	0.978166
	cnn.base.c	2	8.466742	7.450034	1.136470
Loc1	Tile N and K by 8, shift C down, temp out	1	5.892365	3.664158	1.608109
		2	8.581572	8.111660	1.057930
Par1	N loop, dynamic schedule	1	6.277623	0.389050	16.135758
		2	8.194508	0.807715	10.145291

Explanations:

1. Tiling N & K made the most sense to me. They are the loops which are outermost to the array and our datasets are small enough that I think the cache can hold some of the innermost loops. I don't think I did it right though because I didn't get speed up from that alone. I tried tons of tiling schemes, but none seemed to be potent enough to get speed up. However, I did notice that tiling N and K gave speed up when I added a temporary variable for the reduction to out. This makes me think that I didn't tile N and K small enough, but that brings into question whether they are worth tiling at all. If a single loop of n or k struggles to fit in the cache entirely then you should be looking at the loops below. Moving C down in the hierarchy gave me an additional .2 speedup. I think this is because I was able to move the allocations for out and its push back into the output_par to be done once for all iterations of C and not every iteration.
2. Parallelizing the N loop was by far the fastest solution I found in project 1. The interesting thing is that it doesn't give as good of gains here because the loop is tiled now too. I tried adding a reduction over the C loop but I still can't seem to solve that.

I wish we would spend time in class going over tiling and reductions **as a walkthrough**.