Assignment 2 - Programming for Performance - CS377

Name: Estevan Diaz

UTID: etd349

Date: 2/14/2017

Questions

Question 1

For the smallest matrix size, do the L1 and L2 miss rates vary for the different loop-order variants? Does it vary for the larger matrix sizes? Is there any difference in behavior between the different problem sizes? Can you explain the reasons for this behavior?

- For the smallest matrix, the L1 miss rates do not vary significantly. We can attribute this behavior to the likelihood that the L1 cache can accomodate the majority of the space that the smallest matrix takes up. If the L1 cache can accomdate most of the matrix, then the miss rates will fall as most accesses will result in a cache hit.
- For the smallest matrix, the L2 miss rates do vary, with the highest being a 34% and the lowest being 10%. We can attribute this to the small number of loads from the L2 cache, since the L1 cache is doing the majority of the work. Since the cache isn’t warmed up, cache miss rates rise. Temporal/spatial

locality will play a factor, as the row major storage of the matrix will favor different loop permuations more than others.

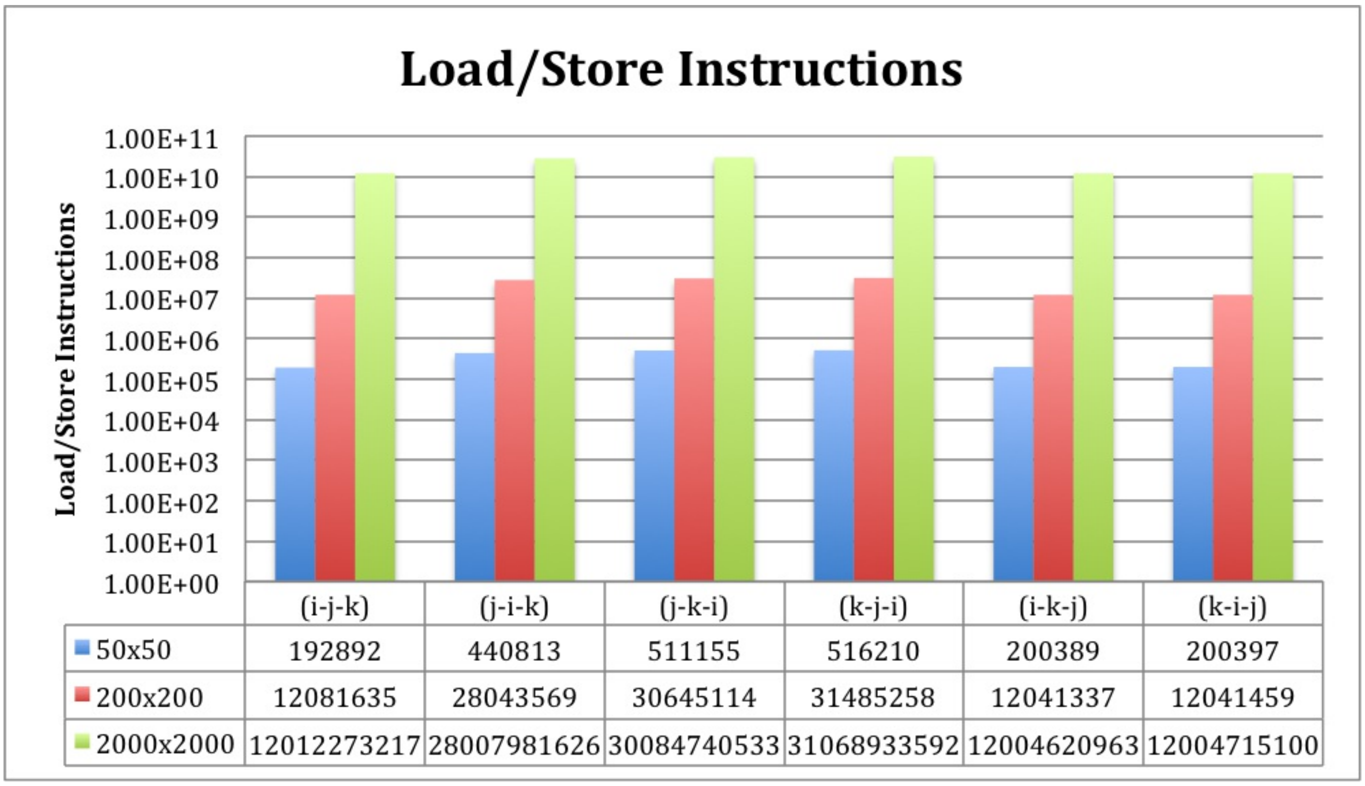
- For the 200x200 matrix, the L2 miss rates also vary. Again, we can note that temporal/spatial locality plays a factor in the spread of the miss rates.
- For the 2000x2000 matrix, we see the most prominent consequence for inefficient loop ordering. The `j-k-i` and `k-j-i` loops have 100% miss rates in the L1 Cache. In this case, the L1 cache is thrashing because as soon as memory is brought into the cache, a new block is needed.
- For the 2000x2000 matrix L2 cache, the miss rate seems to cap of at 85%. For a 2000x2000 double array, the L2 cache is probably operating at full capacity. 85% may very well be the equilibrium that the L2 cache finds between the L1 cache below it and any higher cache (L3) above it.

Question 2

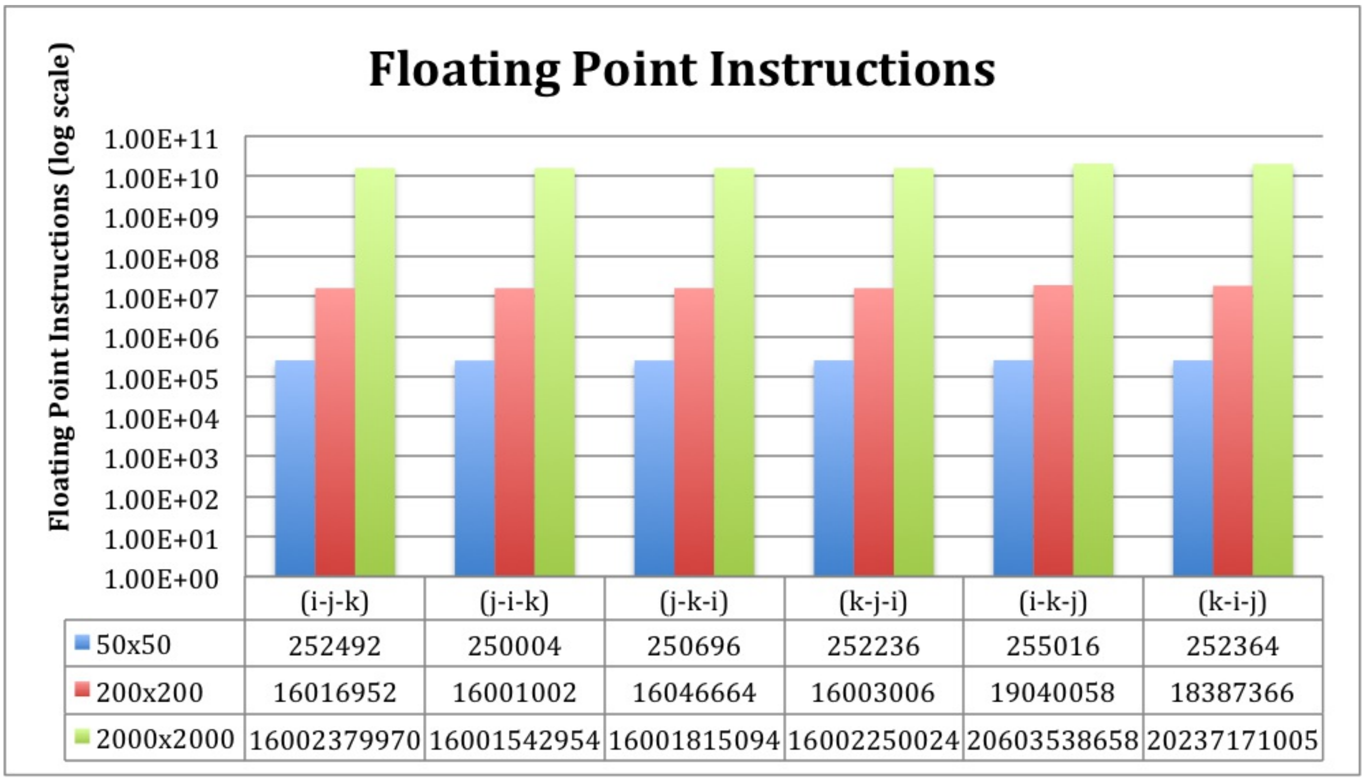
Re-instrument your code by removing PAPI calls and use clock_gettime to measure the execution times for the six versions of MMM and the three matrix sizes specified above. How do your measurements compare to the execution times you obtained from using PAPI to measure the total number of cycles?

- The measurements between the difference in time and the measurements of cycles seems to be 1-to-1. This is intuitive because we ran this job on the `-p serial` partition, meaning that any outside interrupts or context switches wouldn't play a facotr in the `clock_gettime` method of timing our code.

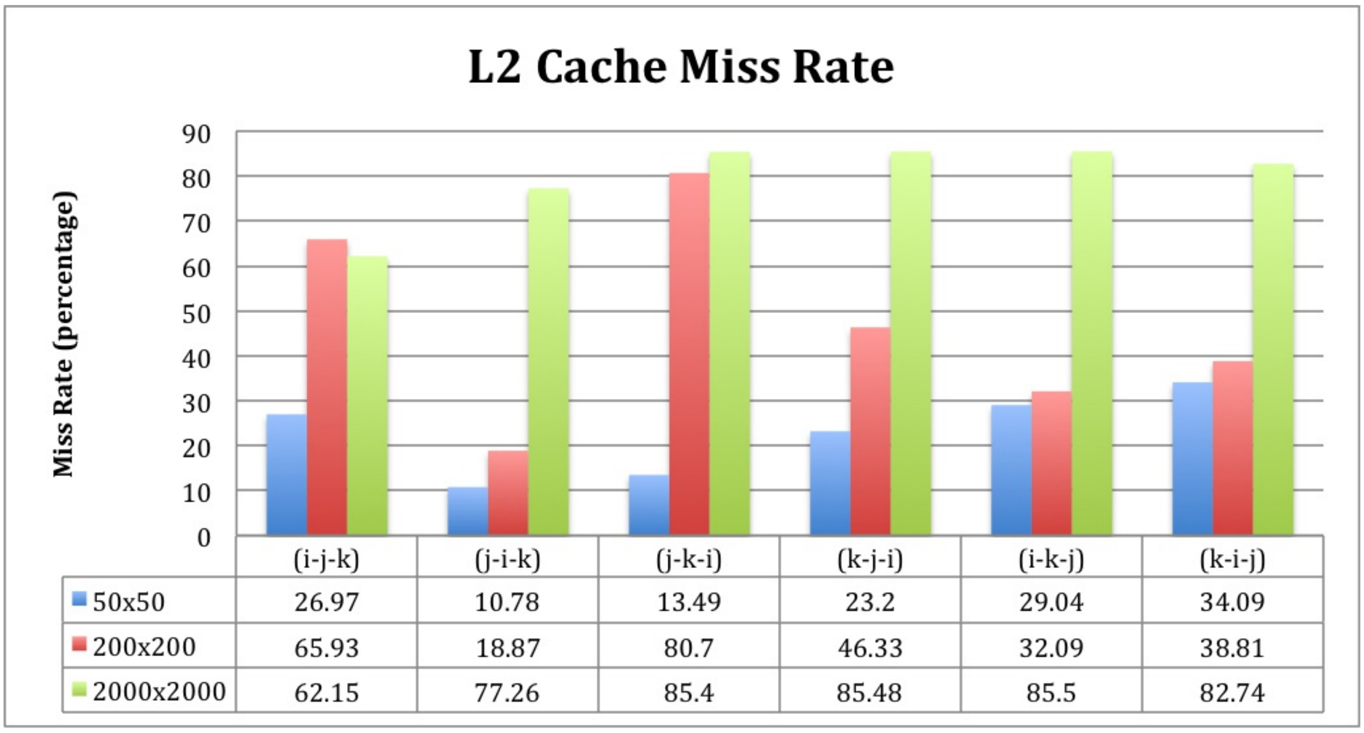
Graphs



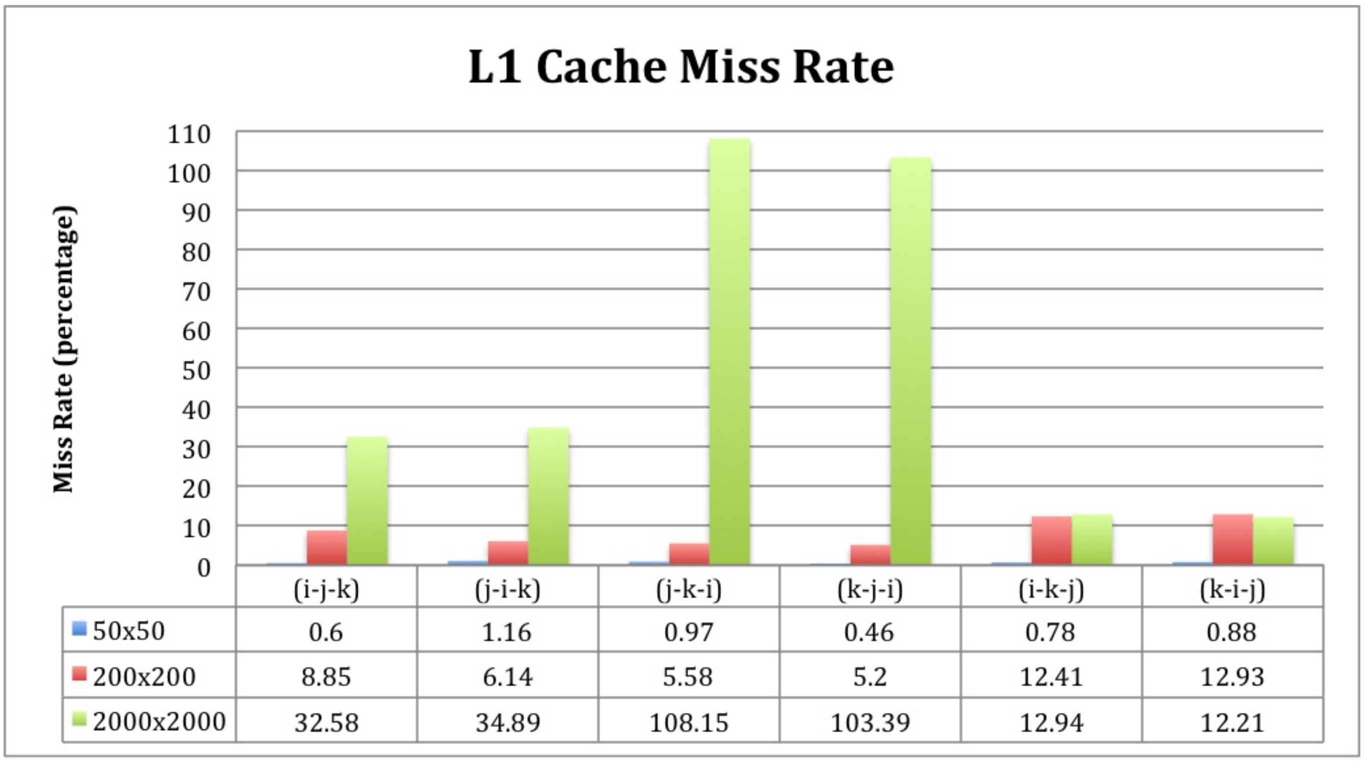
	(i-j-k)	(j-i-k)	(j-k-i)	(k-j-i)	(i-k-j)	(k-i-j)
🟦 50x50	192892	440813	511155	516210	200389	200397
🟥 200x200	12081635	28043569	30645114	31485258	12041337	12041459
🟩 2000x2000	12012273217	28007981626	30084740533	31068933592	12004620963	12004715100



	(i-j-k)	(j-i-k)	(j-k-i)	(k-j-i)	(i-k-j)	(k-i-j)
🟦 50x50	252492	250004	250696	252236	255016	252364
🟥 200x200	16016952	16001002	16046664	16003006	19040058	18387366
🟩 2000x2000	16002379970	16001542954	16001815094	16002250024	20603538658	20237171005



	(i-j-k)	(j-i-k)	(j-k-i)	(k-j-i)	(i-k-j)	(k-i-j)
🟦 50x50	26.97	10.78	13.49	23.2	29.04	34.09
🟥 200x200	65.93	18.87	80.7	46.33	32.09	38.81
🟩 2000x2000	62.15	77.26	85.4	85.48	85.5	82.74



	(i-j-k)	(j-i-k)	(j-k-i)	(k-j-i)	(i-k-j)	(k-i-j)
🟦 50x50	0.6	1.16	0.97	0.46	0.78	0.88
🟥 200x200	8.85	6.14	5.58	5.2	12.41	12.93
🟩 2000x2000	32.58	34.89	108.15	103.39	12.94	12.21

Data

L1 Cache Miss Rate

	50x50	200x200	2000x2000
i-j-k	0.60	.85	32.58

j-i-k	1.16	6.14	34.89
j-k-i	0.97	5.58	108.15
k-j-i	0.46	5.20	103.39
i-k-j	0.78	12.41	12.94
k-i-j	0.88	12.93	12.21

L2 Cache Miss Rate

	50x50	**200x200**	**2000x2000**
i-j-k	26.97	65.93	62.15
j-i-k	10.78	18.81	77.26
j-k-i	13.49	80.70	85.40
k-j-i	23.20	46.33	85.48
i-k-j	29.04	32.09	85.50
k-i-j	34.09	38.81	82.74

Floating Point Operations

	50x50	**200x200**	**2000x2000**
i-j-k	252492	16016952	16002379970
j-i-k	250004	16001002	1600154954
j-k-i	250696	16046664	16001815094
k-j-i	252236	16003006	16002250024
i-k-j	255016	19040058	20603538658

k-i-j	252364	18387366	20237171005

Load/Store Instructions

	50x50	**200x200**	**2000x2000**
i-j-k	192892	12081635	12012273217
j-i-k	440813	29042569	28007891626
j-k-i	511155	20645115	20084740533
k-j-i	516210	31485258	31068922592
i-k-j	200389	12041337	12004620963
k-i-j	200397	12031359	12004715100