Robert Smith

Program 2 - Experiments with Caching

Experimental Design

The matrices were implemented as 1-dimensional short int arrays, and were initialized outside of the timer. A function was created so that the rows & columns could be easily accessed, and the blocking loop was implemented similarly to the example. The timer was begun after the matrix initialization, and ends immediately after the matrix multiplication is completed. Finally, the experiment was run on the Tesla server.

Output

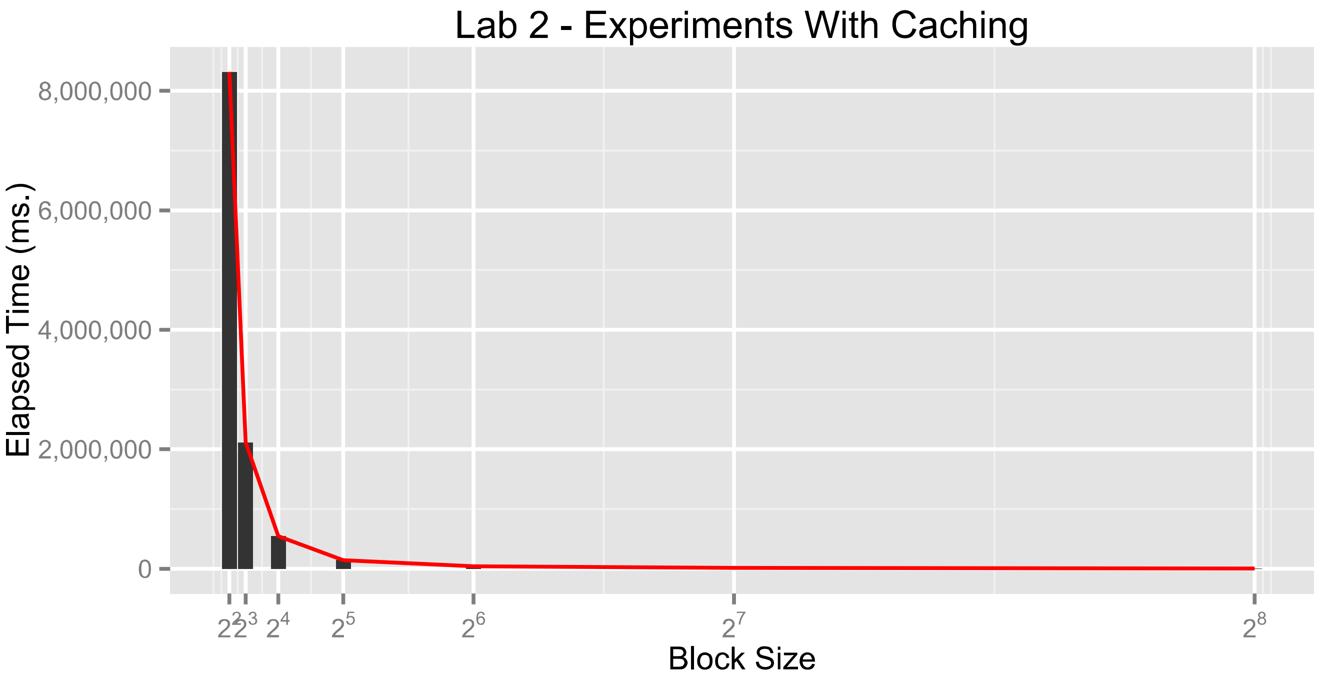
Based on three trials, I found a quite profound & repeatable speedup between Block Sizes of 2^2 and 2^4, with slower speedups in the ranges of 2^5 to 2^8. This is expemlified when we look at the speedup on a logarithmic scale. From the table below, none of the times were outside a certain range so I don't expect that there will be any extraordinary speedup achieved by moving towards larger block sizes.

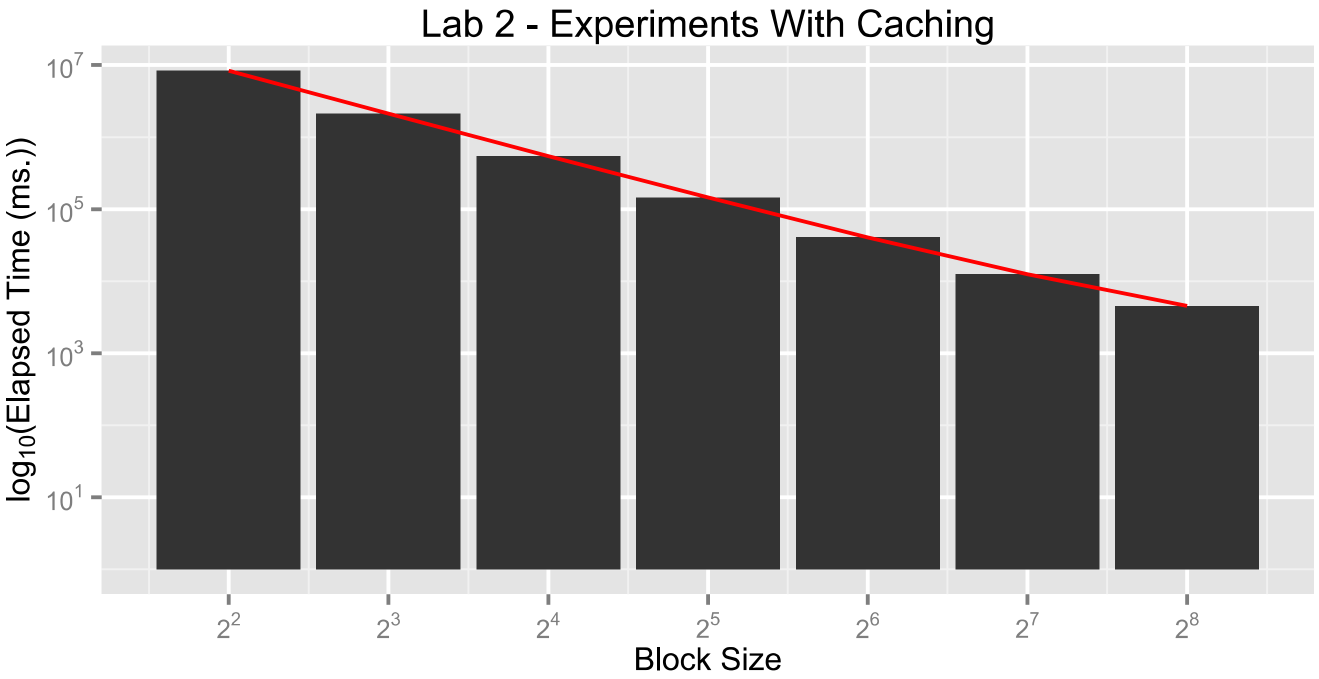
Table of Minimum Elapsed Time

	Block_Size	Elapsed_Time	exp
1	4	8319260.00	1
2	8	2112700.00	2
3	16	544902.00	3
4	32	144702.00	4
5	64	40633.00	5
6	128	12509.90	6
7	256	4520.43	7

Table of Elapsed Times from All Runs

	Block_Size	Run_1	Run_2	Run_3
1	4.00	8319260.00	8334450.00	8319340.00
2	8.00	2112870.00	2116300.00	2112700.00
3	16.00	544926.00	545747.00	544902.00
4	32.00	144890.00	145050.00	144702.00
5	64.00	40633.00	40688.70	40641.60
6	128.00	12509.90	12561.30	12581.50
7	256.00	4530.66	4520.43	4549.03





Conclusions

Based on the trials, I estimate that the cache size is around $2 * 2^9 = 2^{10}$ bytes (as short ints are stored as 2-byte data types), so approximately 1 kilobyte. To find this estimate, we see a fairly linear time trend when we look at the logarithmic value of the time it takes to complete the process which suggests that since we don't see a local minimum somewhere outside the left and right side of the graph that we've approached the cache size but not actually surpassed it.