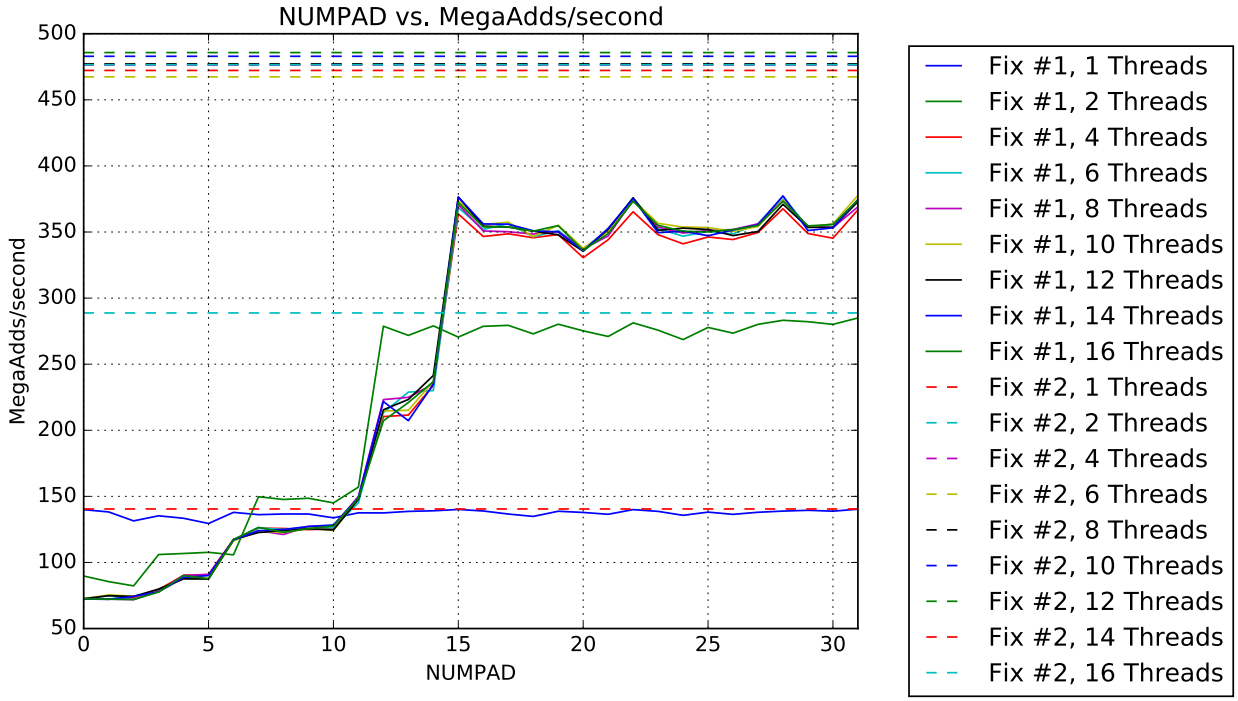
Andrew Alferman
CS 575 Spring 2017
Project #3

1. Tell what machine you ran this on.
    - The machine was a MacBook Pro (Retina, 13-inch, Early 2015) with a 2.9 GHz Intel Core i5 processor running macOS Sierra Version 10.12.4 with 8 GB 1867 MHz DDR3.
2. Create a table with your results.
    - See attached. For sake of completeness, the number of threads went up to 16 instead of 4, and the NUMPAD went up to 31 instead of 15.
3. Draw a graph. The X axis will be NUMPAD. The Y axis will be the performance in whatever units you sensibly choose. There should be at least 6 curves shown together on those axes:
    - 1-3: Using padding with 1, 2, and 4 threads (Fix #1).
    - 4-6: Using a private variable with 1, 2, and 4 threads (Fix #2).

    The graph can be seen below. Several more curves are shown than required because the number of threads went up to 16.



4. What patterns are you seeing in the speeds?
    - With only one thread, Fix 1 is exactly the same speed as Fix 2.

- Computation is slower with two threads than it is with one thread for small values of NUMPAD. The speed using two threads rises slightly around NUMPAD=3, and finally becomes slightly faster than using one thread around NUMPAD=7. The speed using 2 threads rises substantially at NUMPAD=12, and remains relatively constant for all subsequent values of NUMPAD up to 31.
- Computation using four threads is even slower than the speed at 2 threads at small values. The computation speed increases in a less pronounced manner up to NUMPAD=5, and jumps up a little between NUMPAD=6 and NUMPAD=10. Around NUMPAD=12, the four thread speed substantially increases and finally becomes notably faster than the one thread speed. The speed substantially increases again at NUMPAD=15, finally becoming faster than the 2 thread speed. After NUMPAD=15, a few smaller spikes are seen in the data, however the speed remains relatively consistent.
- As the number of threads increases above four, the speeds are the same as the speeds of four threads for each run.
- When more than one thread is used, Fix 2 is faster than Fix 1 in all cases. This effect is small but notable when using two threads, and is larger when using four or more threads.
- The Fix 2 speeds were relatively constant when using four or more threads.

5. Why do you think it is behaving this way?
    - The behavior is similar when using four or more threads because the laptop that was used only has 4 cores. Despite using more threads, the same cores are doing all the work and therefore false sharing isn't an issue beyond what would be seen with four threads.
    - When using only one thread, the speed is constant between Fix 1 and Fix 2 because no false sharing will occur when only using one thread (all work is serial).
    - As explained in the class notes, less padding is needed for two threads than is required for four threads because less padding is required to push data onto two cache lines than is required for four cache lines. When using only 2 threads, only 2 cache lines are needed to prevent false sharing. The substantial performance boost at NUMPAD=12 for two threads is because all the data is finally pushed onto two cache lines at that value. A similar performance boost is seen with four or more threads at NUMPAD=15 because all the data is pushed onto four separate cache lines at that point.
    - Fix 2 is faster than Fix 1 in all cases where more than one thread is used. The reason for this is not clear from the class notes, but it may have to do with the fact that Fix 2 does not require any padding and therefore may make better use of available memory than Fix 1. It may also have to do with optimizations that have been programmed into the compiler, or potentially some kind of trick in the computer's architecture or in the Apple programming.

Table 1: Computation speeds in MegaAdds/second

	1 Thread	2 Threads	4 Threads	6 Threads	8 Threads	10 Threads	12 Threads	14 Threads	16 Threads
Fix #1 - NUMPAD = 0	139.94	89.73	72.61	72.66	72.34	72.54	72.50	72.36	72.47
Fix #1 - NUMPAD = 1	138.22	85.50	72.19	74.82	72.26	75.42	74.85	72.08	72.21
Fix #1 - NUMPAD = 2	131.44	82.27	73.53	72.11	71.96	74.54	74.26	74.33	71.91
Fix #1 - NUMPAD = 3	135.30	105.99	79.47	78.12	78.39	77.97	79.87	77.86	77.60
Fix #1 - NUMPAD = 4	133.42	106.77	90.24	88.03	90.23	88.14	87.58	88.64	90.08
Fix #1 - NUMPAD = 5	129.48	107.66	90.83	87.51	90.93	87.77	87.54	90.10	87.75
Fix #1 - NUMPAD = 6	137.90	105.76	117.39	117.12	117.17	116.41	117.34	117.02	117.22
Fix #1 - NUMPAD = 7	136.15	149.73	125.96	126.18	124.05	123.71	122.65	123.99	126.56
Fix #1 - NUMPAD = 8	136.66	147.63	125.70	124.95	121.16	122.66	124.14	124.92	122.86
Fix #1 - NUMPAD = 9	136.65	148.58	124.82	126.22	127.26	125.81	125.48	127.25	125.54
Fix #1 - NUMPAD = 10	133.75	145.05	125.53	126.22	127.46	124.24	124.62	128.36	127.65
Fix #1 - NUMPAD = 11	137.53	156.99	148.16	145.18	149.94	148.77	147.75	149.01	148.59
Fix #1 - NUMPAD = 12	137.47	278.73	210.36	213.38	223.19	214.54	215.51	221.74	207.13
Fix #1 - NUMPAD = 13	138.64	271.79	211.50	228.87	224.93	215.14	223.31	207.26	221.00
Fix #1 - NUMPAD = 14	139.08	278.99	233.92	230.02	234.96	236.93	241.43	234.77	236.80
Fix #1 - NUMPAD = 15	140.08	270.50	363.66	368.41	370.77	373.57	376.42	376.49	372.28
Fix #1 - NUMPAD = 16	138.94	278.65	346.60	351.98	350.81	355.63	354.45	356.12	353.87
Fix #1 - NUMPAD = 17	136.63	279.38	348.65	356.25	350.20	357.48	353.78	355.89	353.76
Fix #1 - NUMPAD = 18	134.83	272.95	345.68	346.40	348.45	346.63	350.66	350.95	350.65
Fix #1 - NUMPAD = 19	138.73	280.26	348.09	351.05	350.31	355.06	347.89	350.08	354.83
Fix #1 - NUMPAD = 20	137.83	275.23	330.57	337.22	337.23	337.73	335.44	335.71	336.07
Fix #1 - NUMPAD = 21	136.44	270.98	344.15	351.57	347.03	350.84	352.06	352.79	348.86
Fix #1 - NUMPAD = 22	139.97	281.27	365.34	375.80	374.36	374.31	375.87	375.25	373.21
Fix #1 - NUMPAD = 23	138.65	275.72	347.95	352.56	353.82	356.54	351.27	349.50	355.06

Table 1: Computation speeds in MegaAdds/second (continued)

Fix #1 - NUMPAD = 24	135.65	268.63	341.05	346.85	349.20	353.75	352.94	350.67	350.59
Fix #1 - NUMPAD = 25	138.09	277.80	346.23	350.50	351.16	353.41	351.93	347.17	350.12
Fix #1 - NUMPAD = 26	136.39	273.48	344.30	348.60	351.87	350.55	347.31	351.55	351.83
Fix #1 - NUMPAD = 27	137.97	280.14	349.63	356.28	356.18	354.18	350.40	355.52	354.90
Fix #1 - NUMPAD = 28	138.85	283.20	367.57	376.67	374.24	374.30	370.98	377.37	373.33
Fix #1 - NUMPAD = 29	139.45	282.14	348.89	353.56	354.16	353.45	353.50	351.10	354.65
Fix #1 - NUMPAD = 30	138.80	280.12	345.31	356.04	353.31	356.14	353.59	353.20	355.67
Fix #1 - NUMPAD = 31	140.29	284.97	366.68	373.73	369.03	377.73	372.96	374.62	374.08
Fix # 2	140.46	288.77	476.35	467.38	477.17	482.94	485.75	472.24	476.56