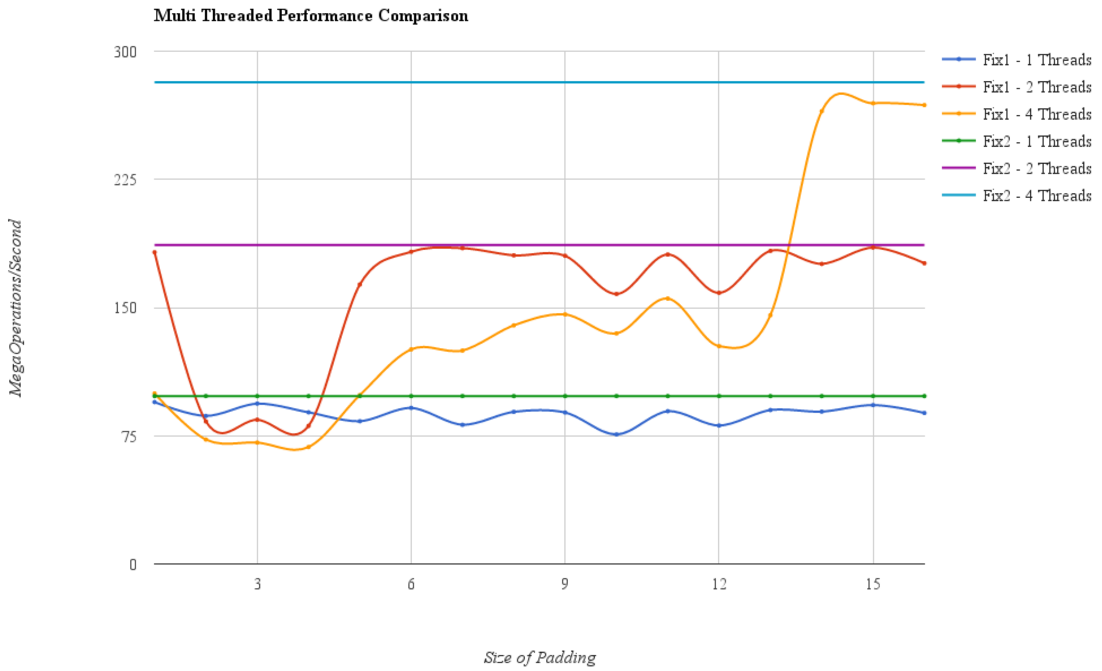
Cody Malick
Project 1
CS 475

1. I ran this on my home desktop, which is running an Intel Sandy Bridge processor with four cores, no hyper threading. It has 8 Gigabytes of RAM.
2. Here is the table of my results:

	Fix1 - 1 Threads	Fix1 - 2 Threads	Fix1 - 4 Threads	Fix2 - 1 Threads	Fix2 - 2 Threads	Fix2 - 4 Threads
1	94.7	182.37	99.77	98.22	186.66	281.83
2	86.6	83.4	72.78	98.22	186.66	281.83
3	93.8	84.4	71	98.22	186.66	281.83
4	88.7	80.76	68.4	98.22	186.66	281.83
5	83.5	163.52	98.78	98.22	186.66	281.83
6	91.3	182.65	125.52	98.22	186.66	281.83
7	81.4	184.75	124.84	98.22	186.66	281.83
8	89	180.55	139.6	98.22	186.66	281.83
9	88.6	180.3	146	98.22	186.66	281.83
10	75.78	157.94	134.9	98.22	186.66	281.83
11	89.39	181.06	155.27	98.22	186.66	281.83
12	80.94	158.6	127.44	98.22	186.66	281.83
13	90.04	183.16	145.57	98.22	186.66	281.83
14	89.12	175.53	264.87	98.22	186.66	281.83
15	92.95	185.13	269.52	98.22	186.66	281.83
16	88.35	175.93	268.46	98.22	186.66	281.83

3. Here is a screen capture of the performance of the different runs. The graph's x axis → size of padding and the y axis → performance in megaoperations/second.



4. With padding, we see a jump in performance after one thread around the fourth padding size, also increasing more for four threads around the fourteenth pad size.

   With fix two, we see that it's actually vastly more effective at increasing performance with the added benefit of not having to manually pad your data structures. I find this method much faster and easier to implement.

5. What's happening here is that the different operations are being spread out over different cache lines, so we see the performance increasing. Because the false sharing is eliminated, we get a nice performance boost after the data is spread out enough.

   With fix two, we can see that it performs very well, vastly better than any of the fix one methods. The reason for this is it completely eliminates cache sharing as a problem. None of the threads will crash over each other trying to write to the same cache line. They simply get different variables to work with. Another added benefit is a reduced size of the structure you're working with because there's no padding array needed.