Miles Van de Wetering

Parallel Project 3

   I ran my code on flip2. I am not entirely sure why my numbers look the way they do. I ran the code three times (at different times of the day) and got pretty much the same results (barring minor fluctuations). The first thing that I notice is that Fix #2 was more efficient across the board for 1 and 2 threads, but that the padding was much more effective when dealing with 4 (and presumably more) threads. I suspect this is due to the convenience of grabbing one cache line at a time, and the fact that modifying that cache line wouldn't invalidate any other thread, obviating the need for private variables completely. It at least met my expectations that the 2 thread variant outperformed the one thread variant at all levels.

   All in all, I think this solidly demonstrates the utility of padding, since we saw great improvement as the padding approached one full cache line in the four threaded version. This matches relatively well with the data which you presented in class. As to the minor fluctuations... I'm definitely blaming the aliens.
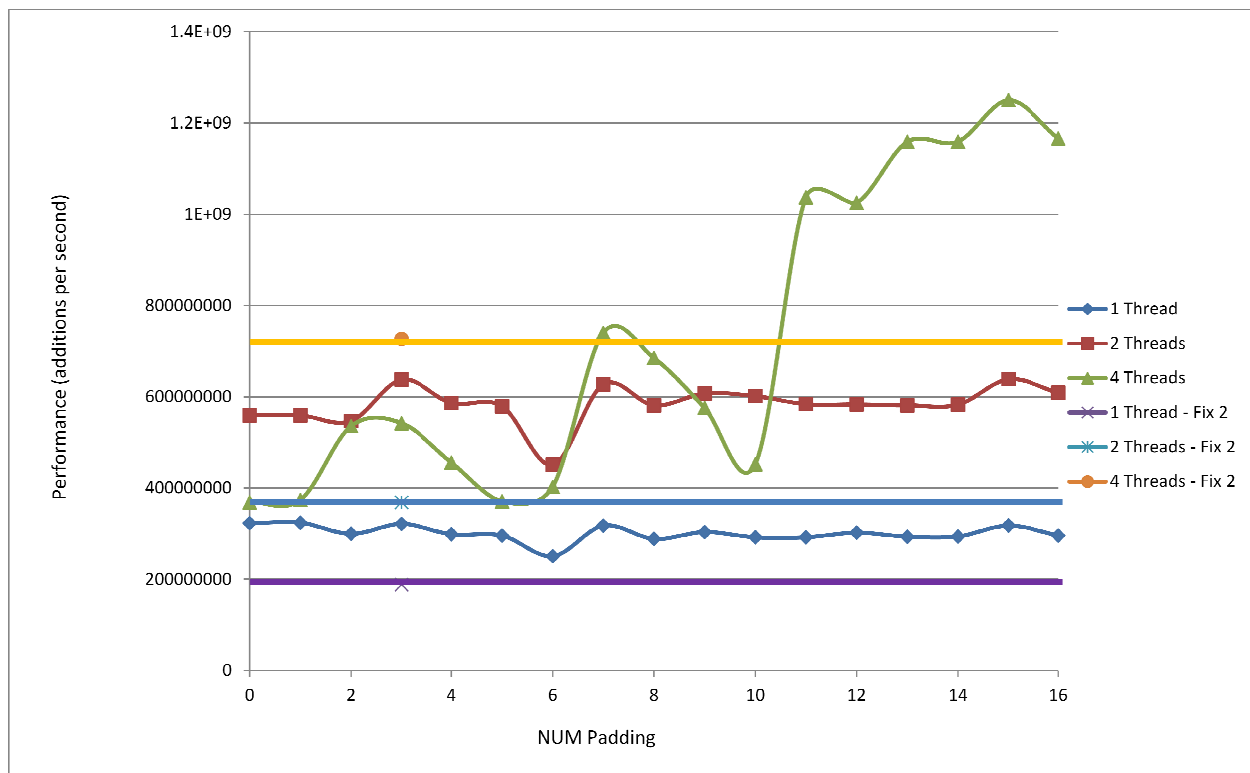
Graph of results:

Table of results:

Performance Data

Padding	1 Thread	2 Threads	4 Threads	1 Thread - Fix 2	2 Threads - Fix 2	4 Threads - Fix 2
0	323056352.3	559608095	368374595.2	1.88E+08	3.69E+08	7.27E+08
1	324000118.8	559495973.4	373667135			
2	299600559.2	546886215.4	535613066.7			
3	321296135.4	637773090.6	542230910			
4	298727297.5	586218031.4	455764437.5			
5	295961658.7	579483793.3	370415247.4			
6	250869848.3	450636576.6	402749369.5			
7	317538431.4	627310480.6	740339189			
8	288661163.2	581344784.3	685089644			
9	303593432.6	607264940.8	575628007.5			
10	292154337.3	601641363.8	452501343.3			
11	292149430.8	583909360.9	1037597525			
12	302270565.3	583449123.1	1025334358			
13	293523473	581531002	1159067840			
14	294070252.6	582745645	1159530208			
15	317669646.4	638642987.2	1250500639			
16	296028722.6	608774889.9	1166333563			