CS 475/575 -- Spring Quarter 2017

Project #3

False Sharing

Professor Mike Bailey

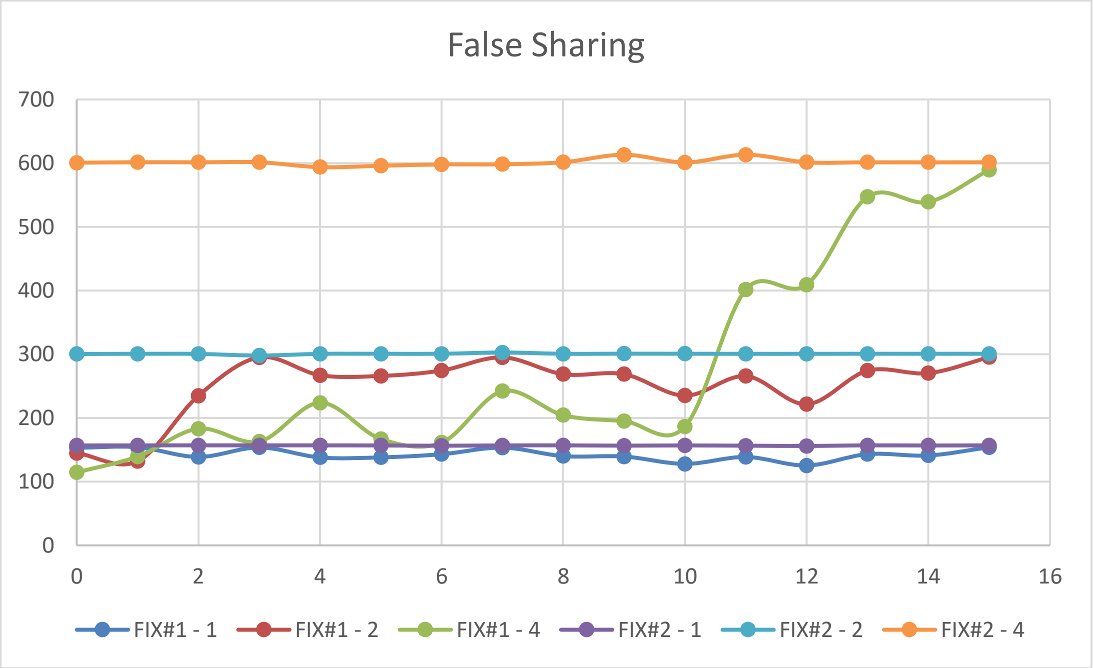
Xiao Tan

I. Runtime environment

In this project, I ran my program on OSU’s server flip.engr.oregonstate.edu. Then I got the following results, and graph.

II. Results and graph

	FIX#1 - 1	FIX#1 - 2	FIX#1 - 4	FIX#2 - 1	FIX#2 - 2	FIX#2 - 4
0	153.17	144.85	114.58	156.87	300.37	600.7
1	154.12	132.03	139.4	156.99	300.74	601.51
2	138.84	234.81	183.18	157.06	300.55	601.36
3	153.63	295.12	162.88	157.01	297.95	601.43
4	138.01	267.11	223.9	157.01	300.63	593.72
5	138.13	266	167.04	156.86	300.78	595.91
6	143.13	274.44	161.36	156.41	300.79	598
7	153.42	295.12	242.08	156.95	302.79	598.25
8	140.11	268.69	204.36	157.01	300.75	601.5
9	139.43	268.7	195	156.57	300.89	613.22
10	127.9	235	186.6	156.87	300.8	601.22
11	138.8	265.7	401.6	156.49	300.68	613.25
12	125.4	221.6	409.2	156.05	300.7	601.33
13	143.2	274.2	547.2	157.02	300.78	601.43
14	141.1	270.3	539.1	156.8	300.75	601.3
15	154	295.3	589.6	157	300.75	601.41



III. Pattern analysis

In my graph, I can see there almost got the same horizontal line in the results, and the performance rising very early for the FIX#1 thread 2, but not got a very performance when the NUMPAD grow higher, FIX#1 thread 4 on contrary, it rises when NUMPAD turned to around 10, and got a higher performance than thread 2. For FIX#2, because it does copy, they got no relative with NUMPAD, so it turns out horizontal line for each thread, and performance grew with thread increasing.

FIX#1 is add NUMPAD to make data skip to another cache line, so it should have different performance relate to NUMPAD increasing, and when threads got data from different cache line, performance must be rising. Moreover, because cache line is 64 bytes, and float number is 4 bytes, so we can assume there are 16 blocks, and when each Array add 5 NUMPAD, there will be some data put in another cache line, so that not cause false sharing, and performance will rise, so is thread 4, it should happen when NUMPAD = 5, 10 and 15. For FIX#2, because it does copy data, so its performance should be increased when add thread number.

Theoretically, in FIX#1 thread 2, it should have changes when NUMPAD increased to 5, however, I always get it rising while NUMPAD = 3, but the rising pattern is right. So, I presume it may be some unknown problem happened in server.