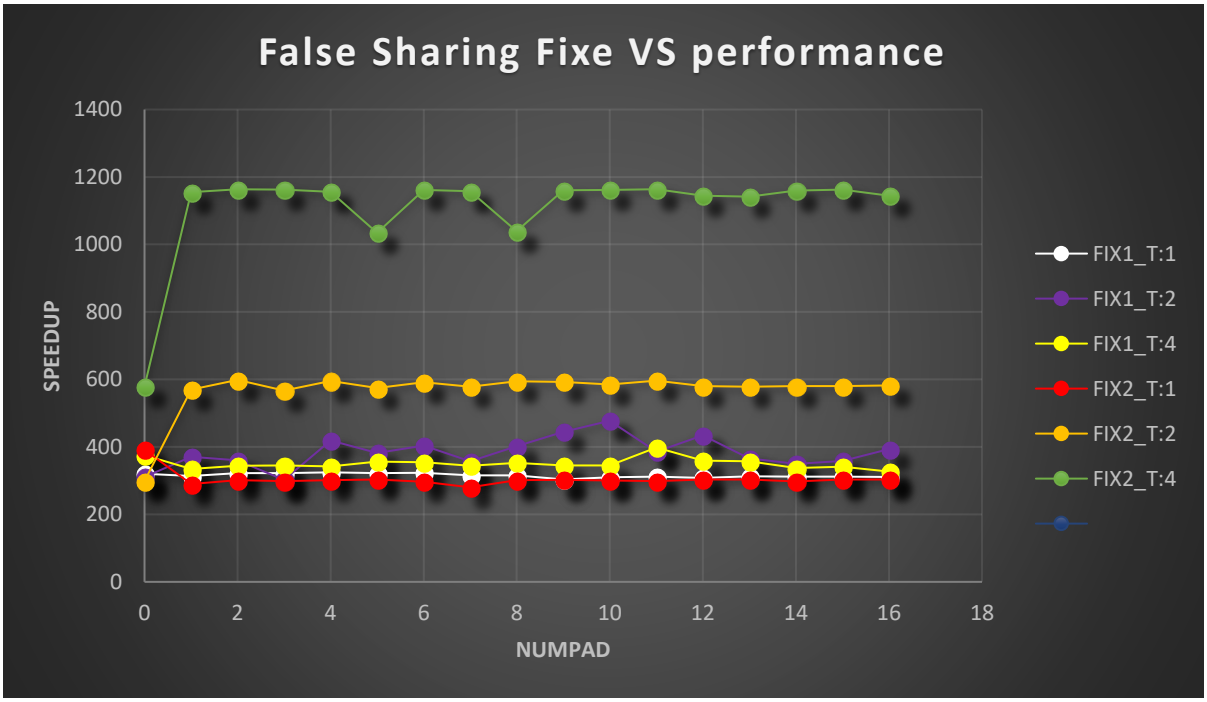
Hanif Khan
CS575 OSU
Project 3 Report (FALSE SHARING)

1. Tell what machine you ran this on.
   I ran this program on two different machine, 1: Flip server
   and 2: my local machine (12GB RAM,2.2GHz, Intel i5, 1TB HDD)
2. Create a table with your results.

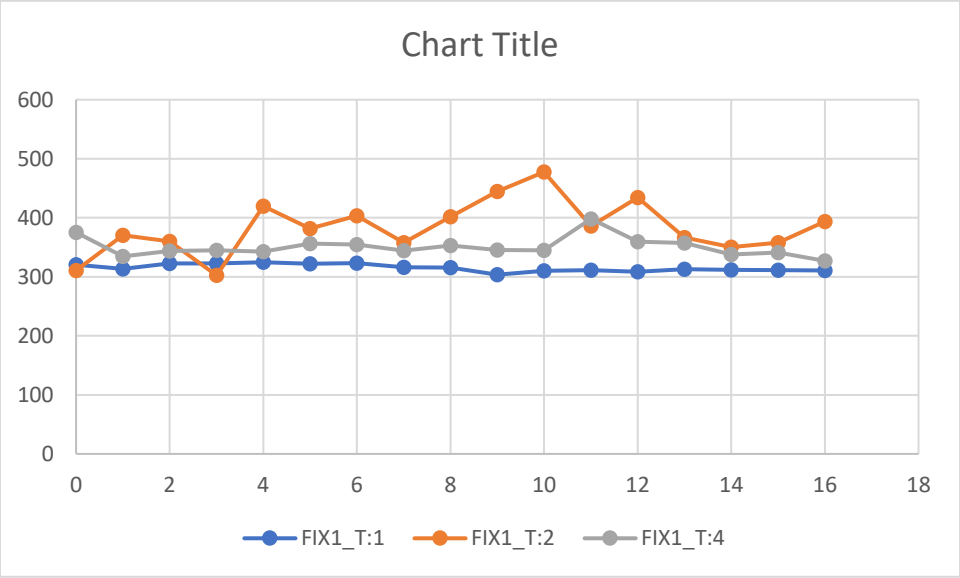
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
FIX1_T:1	320.47	313.7	322.7	322.56	324.62	322.33	323.45	316.12	315.82	303.59	310	311.52	308.52	312.95	312.07	311.53	310.76
FIX1_T:2	310.76	370.27	360.099	302.36	419.414	381.86	403.65	357.86	402.01	444.72	477.44	385.8	434.13	366.35	350.152	358.045	393.44
FIX1_T:4	375.37	334.82	344.01	344.9	342.54	356.39	354.47	344.52	353.04	345.38	344.88	398.2	359.28	357.49	337.65	341.22	327.009
FIX2_T:1	391.6	290.2	301.12	298.22	301.18	303.022	296.36	280.21	301.81	302.19	300.85	298.2	303.1	303.59	298.09	302.9966	303.13
FIX2_T:2	298.68	570.13	598.61	566.93	595.64	574.76	591.74	578.72	594.14	592.21	585.21	596.66	580.1	578.28	580.49	580.53	582.22
FIX2_T:4	579.78	1154.95	1163.51	1162.68	1155.98	1034.85	1161.36	1158.02	1039.75	1160.82	1161.33	1163.85	1144.96	1141.5	1160.22	1162.78	1144.322

Draw a graph. The X axis will be NUM, i.e., the amount of integers used to pad the structure. 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.
4-6: Using a private variable with 1, 2, and 4 threads.
Answer:

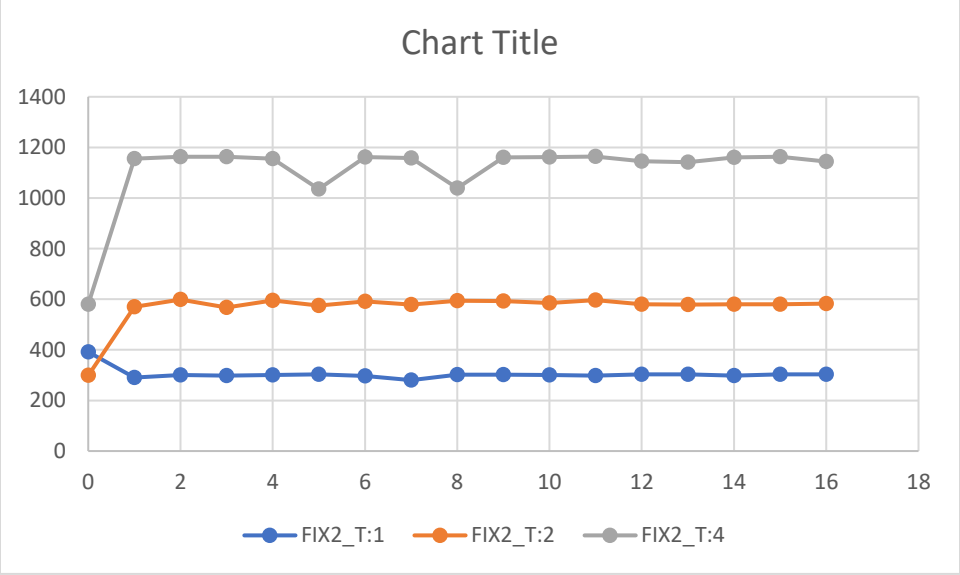


Description: In the figure above, as you can see there are 6 curves. Three of them are for the threads 1, 2 and 4 for the NUMPAD (0 to 16) for the FIX1. Where as other three curves represent for the FIX2 as mentioned in the graph.

Individual graph for fix 1:



For fix2:



3. What patterns are you seeing in the performance?
   Answer:
   For the FIX1: patterns for the performance against the number of NUMPAF is little bit up and down depends on the number of NUMPAD. I have noticed that after 4 threads performance increases suddenly (see the figure for FIX1 and check the performance at NUMPAD 4,8,10,12,16 and compare with its previous performance). I also have observed that after sudden peak performance it again goes down (check the performance at NUMPAD 5,9,13 etc.).
   For the FIX2: Pattern for the performance against the number of NUMPAD is kind of constant. I mean it is almost straight line for the thread 1 and thread 2. However, it is also straight line for the thread 4 but at some points it records some low performance.

4. Why do you think it is behaving this way?
   Answer:
   As explained on the slides in the FIX1 it depends on the NUMPAD and the number of the threads except for thread 1. Since Thread 1 with the NUMPAD 0 in FIX 1 and FIX2 provide the same performance. In the FIX2 all the threads get their private variable and does not depends on numpad and hence provide the same performance. Whereas for the FIX1 cache line is divided using the NUMPAD and shared among the threads to perform the task and hence you can see the peak performance when the NUMPAD is getting larger (except the thread 1). In my scenario, you can observe the same in the graph of thread 4 in the FIX 1.