Bryce Egley

CS475

Project 3 Write Up

1. Tell what machine you ran this on

Flip (Linux) which has 24 processors

1. Create a table with your results.

For all SomeBigNumber = 1000000000

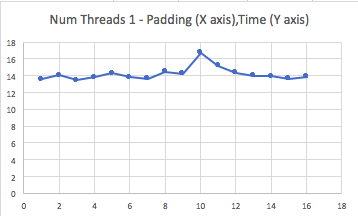
Fix 1

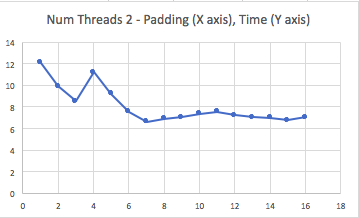
|  |  |  |
| --- | --- | --- |
| NumThreads | Padding | Time |
| 1 | 1 | 13.5631 |
| 1 | 2 | 14.0325 |
| 1 | 3 | 13.4569 |
| 1 | 4 | 13.8215 |
| 1 | 5 | 14.3017 |
| 1 | 6 | 13.7881 |
| 1 | 7 | 13.6414 |
| 1 | 8 | 14.4886 |
| 1 | 9 | 14.223 |
| 1 | 10 | 16.7073 |
| 1 | 11 | 15.177 |
| 1 | 12 | 14.3254 |
| 1 | 13 | 14.0214 |
| 1 | 14 | 13.9329 |
| 1 | 15 | 13.651 |
| 1 | 16 | 13.8702 |
| 2 | 1 | 12.0895 |
| 2 | 2 | 9.92241 |
| 2 | 3 | 8.50226 |
| 2 | 4 | 11.2262 |
| 2 | 5 | 9.21949 |
| 2 | 6 | 7.57356 |
| 2 | 7 | 6.63194 |
| 2 | 8 | 6.92416 |
| 2 | 9 | 7.01967 |
| 2 | 10 | 7.37313 |
| 2 | 11 | 7.53171 |
| 2 | 12 | 7.21495 |
| 2 | 13 | 7.03251 |
| 2 | 14 | 6.98915 |
| 2 | 15 | 6.74949 |
| 2 | 16 | 7.02342 |
| 4 | 1 | 8.24819 |
| 4 | 2 | 10.4736 |
| 4 | 3 | 10.4219 |
| 4 | 4 | 12.3081 |
| 4 | 5 | 7.67386 |
| 4 | 6 | 9.85033 |
| 4 | 7 | 6.16012 |
| 4 | 8 | 6.71257 |
| 4 | 9 | 7.60086 |
| 4 | 10 | 7.13795 |
| 4 | 11 | 6.28258 |
| 4 | 12 | 7.76877 |
| 4 | 13 | 6.67797 |
| 4 | 14 | 6.36726 |
| 4 | 15 | 3.40209 |
| 4 | 16 | 3.6113 |

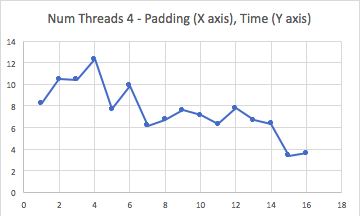
Fix 2

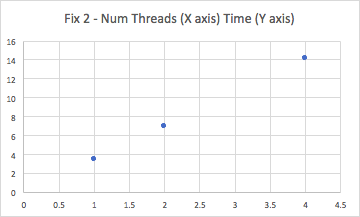
|  |  |
| --- | --- |
| NumThreads | Time |
| 1 | 3.4638 |
| 2 | 6.95983 |
| 4 | 14.1245 |

1. 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.









1. What patterns are you seeing in the performance?

For fix one the pattern I see with padding is that lower padding generally has a lower time, then time reaches a maximum at a padding between 3-7. Then time goes back down as padding increases. With 2 threads though time decrease as padding increases.

For fix two the time increases as the number of threads increases.

1. Why do you think it is behaving this way?

I believe it behaves this way since with false sharing it will cause a time increase but will be fixed with more padding. This is why we get the patterns of the graphs increasing in time at first then decreasing.