Tim Giel

Evan Drewry

PPPP Project Data Discussion

The performance of our code achieved significant speedups in multiple areas, but most importantly, in the total time it took to complete. We also saw speedups in the Map and Reduce steps because these were the only parts of the code that were parallelizable. The Shuffle and Collect steps were very similar in their times due to the fact that we could not parallelize them.

The following graph and chart shows our data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 Async | 2 Asyncs | 4 Asyncs | 8 Asyncs | 16 Asyncs |
| Map | 3778 | 2355 | 1288 | 723 | 623 |
| Shuffle | 287 | 290 | 274 | 277 | 323 |
| Reduce | 422 | 428 | 214 | 145 | 128 |
| Collect | 12 | 17 | 26 | 28 | 23 |
| Total | 4499 | 3092 | 1802 | 1173 | 1100 |