Parallel vs Serial Matrix Vector Multiplication using OpenMP – Basil Lin

12/6/2018

ECE 6730

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem Size** | **Serial Time** | **Parallel Time** | **Speedup** |
| 1000 | 0.006146 | 0.00531 | 1.157439 |
| 1500 | 0.01431 | 0.007297 | 1.96108 |
| 2500 | 0.0375 | 0.009261 | 4.049239 |
| 3500 | 0.07473 | 0.01249 | 5.983187 |
| 5000 | 0.1418 | 0.01939 | 7.313048 |
| 7500 | 0.3241 | 0.03449 | 9.396927 |
| 10000 | 0.5784 | 0.05349 | 10.81324 |
| 20000 | 2.159 | 0.1777 | 12.14969 |

Table 1: Problem Sizes and Times

Figure 1: Execution Time for Serial and Parallel Code

Figure 2: Parallel Speedup in Relation to Problem Size

As seen from the above table and accompanying graphs, parallelizing the matrix-vector multiplication algorithm resulted in a significant speedup. Additionally, as the problem size increased, speedup also increased, although in a logarithmic manner, as shown in the parallel speedup graph. A minimum matrix size of around 1000 is needed in order to witness any speedups, and as shown in the table, a problem size of 1000 leads to a speedup of about 1.16. Problem sizes significantly below 1000 will result in little to no speedup, as the extra overhead of parallel code will often make it run slower than serial code.