|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number Of Cores | Number Of Thread | MatrixSize | Serial Execution time | Parallel Execution time | Speed UP |
| 1 | 2 | 64 | 1549 | 1866 | 0.830118 |
| 2 | 4 | 128 | 11899 | 7071 | 1.682789 |
| 3 | 9 | 256 | 93664 | 37839 | 2.47533 |
| 4 | 12 | 512 | 745933 | 218439 | 3.414834 |
| 32 | 200 | 1024 | 6209086 | 341269 | 18.19411 |
| 32 | 1280 | 2048 | 49722567 | 2821954 | 17.61991 |
| 32 | 2560 | 4096 | 401824448 | 27092697 | 14.83147 |

I have changed my logic from the Assignment 1, as it follow recursive method which consume more time than the for loop and moreover, to the max only four thread can be able to use. The solution that I presented here for HW5 is using for loop and arranges the matrix as a single vector in a order such that it reduce the memory access time. This logic is referred from the internet.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number Of Node | Number Of Thread | MatrixSize | Serial Execution time | |  | | --- | | Distributed Execution time | | Speed UP |
| 1 | 2 | 64 | 1549 | 1866 | 0.830118 |
| 2 | 4 | 128 | 11899 | 7071 | 1.682789 |
| 3 | 9 | 256 | 93664 | 37839 | 2.47533 |
| 4 | 12 | 512 | 745933 | 218439 | 3.414834 |
| 32 | 200 | 1024 | 6209086 | 341269 | 18.19411 |
| 32 | 1280 | 2048 | 49722567 | 2821954 | 17.61991 |
| 32 | 2560 | 4096 | 401824448 | 27092697 | 14.83147 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number Of Node | Number Of Thread | MatrixSize | Serial Execution time | |  | | --- | | Distributed Execution time | | Speed UP |
|  |  |  |  |  |  |
| 2 | 2 | 512 | 745933 | 866188 | 0.861168 |
| 3 | 2 | 512 | 745933 | 587854 | 1.268909 |
| 4 | 2 | 512 | 745933 | 548490 | 1.359976 |
| 5 | 2 | 512 | 745933 | 580634 | 1.284687 |
| 6 | 2 | 512 | 745933 | 561840 | 1.327661 |
| 7 | 2 | 512 | 745933 | 589158 | 1.2661 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number Of Node | Number Of Thread | MatrixSize | Serial Execution time | |  | | --- | | Distributed Execution time **(On UAlbany Cluster)** | | Speed UP |
| 2 | 2 | 64 | 1549 | 10534 | 0.147048 |
| 2 | 2 | 128 | 11899 | 36735 | 0.323915 |
| 2 | 2 | 256 | 93664 | 125739 | 0.744908 |
| 4 | 2 | 512 | 745933 | 564500 | 1.321405 |
| 4 | 2 | 1024 | 6209086 | 2702689 | 2.297373 |
| 4 | 2 | 2048 | 49722567 | 14362392 | 3.461998 |
| 4 | 2 | 4096 | 401824448 | 95905832 | 4.189781 |
|  |  | 5120 |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number Of Node | Number Of Thread | MatrixSize | Serial Execution time | |  | | --- | | Distributed Execution time **(on Amazon EC2)** | | Speed UP |
| 2 | 2 | 64 | 1549 | 7413 | 0.208957 |
| 2 | 2 | 128 | 11899 | 27330 | 0.435382 |
| 2 | 2 | 256 | 93664 | 122763 | 0.762966 |
| 4 | 2 | 512 | 745933 | 537429 | 1.387966 |
| 4 | 2 | 1024 | 6209086 | 2663099 | 2.331527 |
| 4 | 2 | 2048 | 49722567 | 14570805 | 3.412479 |
| 4 | 2 | 4096 | 401824448 | 91520573 | 4.390537 |
|  |  | 4500 |  |  |  |
|  |  |  |  |  |  |