Assignment No. 2

CS476/576-01\_SPG14

Type of CPU: 4-core

* 10240 sized Square Matrix addition using Static Work Division

|  |  |
| --- | --- |
| Splits | Running Time (s) |
| 2 | 13.3770287 |
| 4 | 10.9893684 |
| 8 | 10.9079349 |

* 10240 sized Square Matrix addition using Recursive Divide and Conquer

|  |  |
| --- | --- |
| Splits | Running Time (s) |
| 2 | 14.2965037 |
| 4 | 11.3551396 |
| 8 | 11.2749201 |

* 2000 sized Square Matrix multiplication using Static Work Division

|  |  |
| --- | --- |
| Splits | Running Time (s) |
| 2 | 289.650013 |
| 4 | 182.424124 |
| 8 | 177.592886 |

* 2000 sized Square Matrix multiplication using recursive Divide and Conquer

|  |  |
| --- | --- |
| Splits | Running Time (s) |
| 2 | 285.207755 |
| 4 | 170.20481 |
| 8 | 167.908238 |

**Analysis:**

Matrix addition using static work division:

* The running time gets reduced fairly on increasing the number of splits from 2 to 4.
* However, on increasing the number of splits from 4 to 8, the running time gets reduced by only a very small margin.
* The best running time is obtained with 8 splits.

Matrix addition using recursive Divide & Conquer:

* As with static work division case, the running time gets reduced fairly on increasing the number of splits from 2 to 4.
* Again, increasing the number of splits from 4 to 8 affects the running time to decrease by just short of one second.
* The best running time is obtained with 8 splits.
* Overall, the square matrix addition using recursive divide and conquer takes more time than using static work division for all three splits.

Matrix multiplication using static work division:

* There is a substantial reduction in running time on increasing the number of splits from 2 to 4.
* Increasing the splits from 4 to 8 does not reduce the running time much as compared to the previous case.
* The best running time is obtained with 8 splits.
* From the running times we observed for 2000 size matrix, the time to observe a 10240 size matrix would have been huge.

Matrix multiplication using recursive Divide & Conquer:

* As with static work division case, the running time gets reduced substantially on increasing the number of splits from 2 to 4.
* Again, increasing the number of splits from 4 to 8 affects the running time to decrease by little over two seconds.
* The best running time is obtained with 8 splits.
* Overall, the square matrix multiplication using recursive divide and conquer takes less time than using static work division for all three splits.