|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TIME In milliseconds(ms) | | Personal Computer (8 processors) | | | | Park  (4 processors) | | | | | Byrd  (8 processors) | | | |
| Serial | | Many  Threads | Row  Threads | Serial | Many  Threads | | Row  Threads | | Serial | Many  Threads | | Row  Threads |
| Small  Matrices  2x3 &3x4 | R1 | 0 | | 1 | 1 | 0 | 2 | | 1 | | 0 | 2 | | 1 |
| R2 | 0 | | 1 | 0 | 0 | 3 | | 1 | | 0 | 2 | | 1 |
| R3 | 0 | | 1 | 1 | 0 | 3 | | 1 | | 0 | 2 | | 1 |
| AVG | 0 | | 1 | 0.33 | 0 | 2.67 | | 1 | | 0 | 2 | | 1 |
|  | | | | | | | | | | | | | | |
| Medium  Matrices  60x70 & 70x80 | R1 | 3 | 421 | | 10 | 8 | 526 | | 19 | | 12 | 829 | | 19 |
| R2 | 5 | 430 | | 11 | 7 | 526 | | 18 | | 12 | 854 | | 16 |
| R3 | 4 | 424 | | 11 | 8 | 538 | | 18 | | 12 | 827 | | 16 |
| AVG | 4 | 425 | | 10.33 | 7.67 | 530 | | 18.33 | | 12 | 836.67 | | 17 |
|  | | | | | | | | | | | | | | |
| Large  Matrices  1000x2000 & 2000x3000 | R1 | 147217 | | Out of heap | 132049 | 306966 | | 595043 | | 301093 | 47836 | | 511665 | 464221 |
| R2 | 146387 | | Out of heap | 131904 | 300331 | | 592530 | | 304741 | 49375 | | 512899 | 47282 |
| R3 | 146012 | | Out of heap | 131856 | 307291 | | 594803 | | 308347 | 50177 | | 513098 | 48731 |
| AVG | 146538.67 | | No Result | 131936.33 | 304862.67 | | 594125.33 | | 304727 | 49129.33 | | 521554 | 47478 |
|  | | | | | | | | | | | | | | |
| Conclusion:  From the result of the runtime, it is obvious that ManyThreads is the wrost method for matrix multiplication. Due to the powerful single processor, Serial method is always the fastest, until facing with large Matrices multiplication. For large matrices multiplication, the RowThreads method becomes the fastest in these three.  After all these testing, it is clearly that parallel strategy is not that many more thread increase the speed of the runtime. In large matrices, ManyThread has 3 milions thread, and RowThread has only 1000, but it is clearly that RowTheads runs much more faster that ManyThreads, due to the limit of single processor.  Also ,in this result, it is clear that my processor’s unit speed is faster that byrd,but the full calculation capacity is lower than byrd.(My CPU shows that the heap is out, too many threads for my CPU. | | | | | | | | | | | | | | |