Mean Thread

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| thread | timing (ms) | | | | | 20% trimmed mean |
| 1 | 11 | 11 | 12 | 12 | 13 | 11.667 |
| 2 | 18 | 19 | 19 | 19 | 20 | 19.000 |
| 4 | 21 | 22 | 22 | 24 | 28 | 22.667 |
| 8 | 29 | 44 | 44 | 47 | 48 | 45.000 |
| 16 | 39 | 43 | 44 | 46 | 75 | 44.333 |
| 32 | 31 | 32 | 33 | 38 | 81 | 34.333 |
| 64 | 73 | 73 | 75 | 80 | 112 | 76.000 |
| 128 | 80 | 94 | 94 | 95 | 95 | 94.333 |
| 256 | 86 | 91 | 99 | 102 | 107 | 97.333 |
| 512 | 115 | 119 | 121 | 128 | 129 | 122.667 |
| 1024 | 179 | 189 | 192 | 195 | 199 | 192.000 |
| 2048 | 314 | 327 | 330 | 333 | 334 | 330.000 |

Screenshots of the output of MeanThread programs are included in the annex respectively.

Runtime of the mean threads was measured without printing the temporal means of the individual threads. The program was ran at the specified thread count 5 times and the 20% trimmed mean was used to plot the graphs.

Runtime increases exponentially with the number of threads used. This could be due to the overhead from starting multiple threads. The runtime of the threads can also be affected by CPU resources or RAM of the computer).

The variance amongst runtimes of the same threads can be attributed to the fact that the programs were ran for the first time, so more time is needed to load the data into the cache.

Median Thread

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| thread | timing (ms) | | | | | 20% trimmed mean |
| 1 | 472 | 472 | 479 | 488 | 494 | 479.667 |
| 2 | 364 | 367 | 377 | 383 | 399 | 375.667 |
| 4 | 283 | 329 | 330 | 336 | 339 | 331.667 |
| 8 | 272 | 331 | 332 | 343 | 344 | 335.333 |
| 16 | 307 | 313 | 319 | 321 | 328 | 317.667 |
| 32 | 326 | 328 | 330 | 337 | 382 | 331.667 |
| 64 | 352 | 356 | 365 | 365 | 518 | 362.000 |
| 128 | 413 | 436 | 443 | 449 | 502 | 442.667 |
| 256 | 462 | 477 | 482 | 521 | 521 | 493.333 |
| 512 | 511 | 551 | 561 | 575 | 607 | 562.333 |
| 1024 | 612 | 631 | 636 | 640 | 648 | 635.667 |
| 2048 | 847 | 848 | 870 | 877 | 897 | 865.000 |

Screenshots of the output of MedianThread program is included in the annex. Runtime of the median thread was measured without printing out the entire sorted arrays. As with the mean thread the runtime of each thread count was measured 5 times and the 20% trimmed mean used to plot the graph above.

Runtime initially decreased before increasing exponentially after beyond 16 thread counts. This could be because multi thread programming for calculating the median is initially effective however beyond 16 threads, there began to be significant overhead from running and starting the threads, resulting in the exponential increase.

Runtime could also be affected by CPU resources and Ram available.

Annex

Output from MeanThrea.java



Output from Median Thread program

