**Spin Lock Comparison**

Performance comparison of Spin Lock implementation - On Intel core i-7 processor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Threads** | **TAS** | **TTAS** | **BackOff** | **CLH** |
| 5 | 21 | 40 | 3 | 4 |
| 10 | 9 | 18 | 2 | 1 |
| 15 | 10 | 4 | 2 | 2 |
| 20 | 5 | 7 | 3 | 3 |
| 25 | 5 | 8 | 3 | 3 |
| 30 | 8 | 23 | 4 | 7 |
| 35 | 6 | 10 | 5 | 5 |
| 40 | 7 | 18 | 5 | 6 |
| 45 | 11 | 23 | 6 | 24 |
| 50 | 9 | 12 | 7 | 7 |
| 55 | 13 | 15 | 7 | 8 |
| 60 | 12 | 23 | 8 | 8 |
| 65 | 13 | 16 | 20 | 9 |
| 70 | 14 | 18 | 9 | 10 |
| 75 | 16 | 16 | 11 | 10 |
| 80 | 19 | 29 | 11 | 11 |
| 85 | 21 | 20 | 14 | 11 |
| 90 | 21 | 20 | 14 | 15 |
| 95 | 21 | 25 | 13 | 14 |
| 100 | 21 | 30 | 17 | 17 |

Backoff Lock: optimal values for minDelay=4, maxDelay=200;

1. **TAS Lock** - TAS based implementation does not scale up for large no of threads. It gives worst performance compared to others lock implementation.
2. **TTAS Lock -** TTAS based implementation is better than TAS. But in case of high contentionthis do not perform well.
3. **Backoff Lock -** This lock requires performance tuning based on underline architecture and no of threads i.e we need to find out the optimal values for minDelay and maxDelay.
4. **CLH Lock -** Performance of CLH based locking is better than all other implementation.experiment shows that this is indeed true as it perform equally well or better that Backoff lock