Vipul Sharma
B17069
07-10-2020

CS592 Reactive Design Patterns Assignment A1

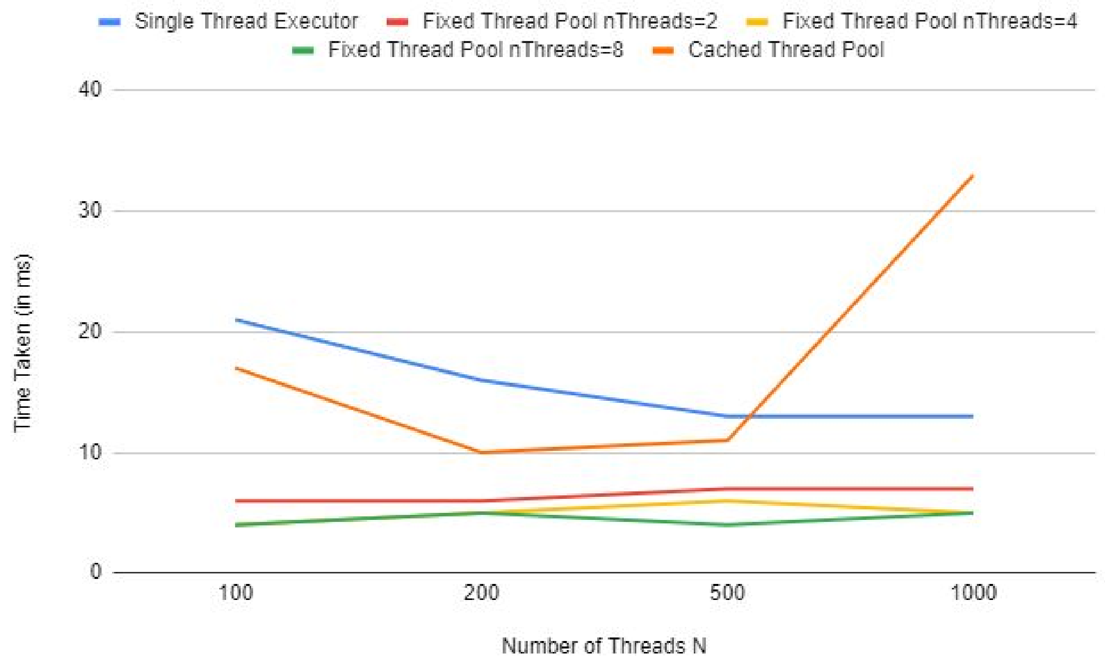
Aim-

a. To explore and get familiar with various mechanisms to manage a thread pool in Java using Executor framework.
b. Write a program that computes all primes in the range 1..MAX, where MAX is a large positive integer. Divide the range into N subranges and create one thread for each subrange; N should again be large, say varying between 100 − 1000

Observations-

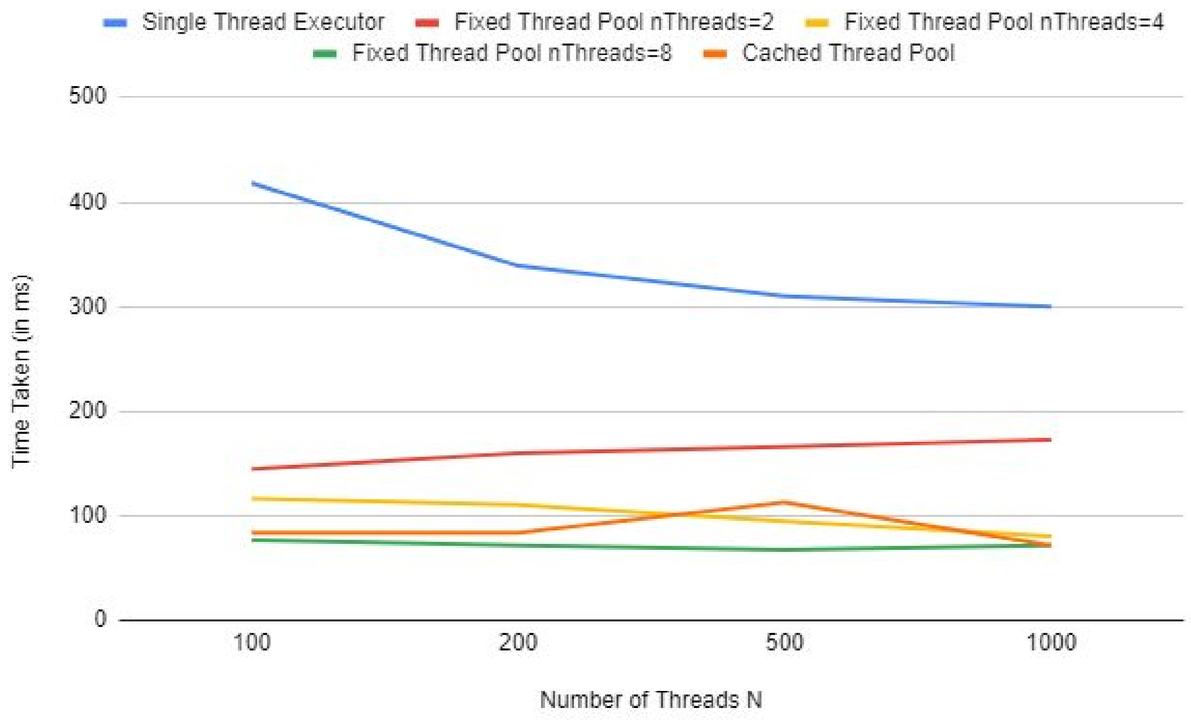
For MAX = 100000

Number of Threads/ Subranges N	Single Thread Executor	Fixed Thread Pool nThreads=2	Fixed Thread Pool nThreads=4	Fixed Thread Pool nThreads=8	Cached Thread Pool
100	21	6	4	4	17
200	16	6	5	5	10
500	13	7	6	4	11
1000	13	7	5	5	33



For MAX = 1000000

Number of Threads/ Subranges N	**Single Thread Executor**	**Fixed Thread Pool nThreads=2**	**Fixed Thread Pool nThreads=4**	**Fixed Thread Pool nThreads=8**	**Cached Thread Pool**
100	418	145	117	77	84
200	339	160	111	72	84
500	310	166	95	68	113
1000	300	173	81	72	72



Note - All time shown is in ms.

Conclusions-

1. The difference between the three types of threads we used are -
   - Single Threaded Pool: Keeps only one thread executing one task at a time.
   - Fixed Thread Pool: Limits the maximum number of threads while the additional tasks wait in a queue.
   - Cached Thread Pool: Keeps a number of alive threads and creates new ones as needed.

2. Best speedup is achieved in Fixed Thread Pool with nThreads = 8 case. This is because there is a balance in this case. Not a lot of time is consumed in performing forks and joins and also the work is divided between the threads in the most optimized way.

3. In some cases for Cached Thread Pool, if a lot of threads are created, this results in more time being consumed for forking and joining these threads. This results in getting poor results for some cases. But, in most cases, there is a significant speedup in comparison to Single Thread Executor Case. These results are much more significantly seen in the MAX=1000000 case.

4. Cached Thread Pool is more suited for applications that require to execute multiple short-lived tasks concurrently. In this experiment, each thread was given a subrange of a significant size to find prime numbers from. So, it may be the case that the tasks were not short-lived to that extent and thus, Cached Thread Pool speedup was slower than some Fixed Thread Pool cases.