Chp 7 - Exercise 2

We are going to use threads to try and speed up a simple application.
Write a serial application that will sum up all the values from 1 to n.
    TIP: You’ll want to use a long to store the result.
Time it for the following values of n: 1 million, 100 million, 1 billion.
Write a threaded version that will create 8 threads, assign part of the summation to each thread and then combine each thread’s result to get the total.
Compare the result of this version to the serial version to make sure it is functioning correctly.
Time it for the same values of n as you did the serial version.
What levels of speed up did you get? Did it match what you would have expected?
Grading Rubric
_____ (1 Point) Serial Version of the code
_____ (2 Points) Parallel Version of the code
_____ (2 Point) Test function for each version of the code
_____ (3 Points) Stated an expectation of results and what results you got.

Serial code

```
public class SerialSummation {
    public static void main(String[] args) {
        long[] ns = {1_000_000L, 100_000_000L, 1_000_000_000L};

        for (long n : ns) {
            long sum = 0;
            long startTime = System.nanoTime();

            for (long i = 1; i <= n; i++) {
                sum += i;
            }

            long endTime = System.nanoTime();
            System.out.println("Sum: " + sum + ", Time: " +
(endTime - startTime) / 1_000_000 + " ms for n = " + n);
        }
    }
}
```

Output of serial:

```
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>  &
```

```
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'SerialSummation'
Sum: 500000500000, Time: 2 ms for n = 1000000
Sum: 5000000050000000, Time: 25 ms for n = 100000000
Sum: 500000000500000000, Time: 486 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'SerialSummation'
Sum: 500000500000, Time: 2 ms for n = 1000000
Sum: 5000000050000000, Time: 25 ms for n = 100000000
Sum: 500000000500000000, Time: 496 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'SerialSummation'
Sum: 500000500000, Time: 2 ms for n = 1000000
Sum: 5000000050000000, Time: 25 ms for n = 100000000
Sum: 500000000500000000, Time: 489 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'SerialSummation'
Sum: 500000500000, Time: 2 ms for n = 1000000
Sum: 5000000050000000, Time: 25 ms for n = 100000000
Sum: 500000000500000000, Time: 487 ms for n = 1000000000
```

```
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'SerialSummation'
Sum: 500000500000, Time: 2 ms for n = 1000000
Sum: 5000000050000000, Time: 25 ms for n = 100000000
Sum: 500000000500000000, Time: 502 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
```

Threaded

```
class SummationThread extends Thread {
    private long start, end, sum;

    public SummationThread(long start, long end) {
        this.start = start;
        this.end = end;
    }

    public void run() {
        sum = 0;
        for (long i = start; i <= end; i++) {
            sum += i;
        }
    }

    public long getSum() {
        return sum;
    }
}

public class ThreadedSummation {
    public static void main(String[] args) {
        long[] ns = {1_000_000L, 100_000_000L, 1_000_000_000L};

        for (long n : ns) {
            long totalSum = 0;
            SummationThread[] threads = new SummationThread[8];
            long startTime = System.nanoTime();
            // Divide the summation range among 8 threads
            long range = n / 8;
            for (int i = 0; i < 8; i++) {
                long start = i * range + 1;
                long end = (i == 7) ? n : (i + 1) * range;
                threads[i] = new SummationThread(start, end);
                threads[i].start();
            }

            // Combine results from all threads
            for (int i = 0; i < 8; i++) {
                try {
                    threads[i].join();
                    totalSum += threads[i].getSum();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }

            long endTime = System.nanoTime();
            System.out.println("Total Sum: " + totalSum + ", 
Time: " + (endTime - startTime) / 1_000_000 + " ms for n = " + 
n);
        }
    }
}
```

Output of threaded:

```
PS G:\Other computers\My MacBook Air\School\Shippensburg 
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>  & 
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview' 
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'ThreadedSummation'
Total Sum: 500000500000, Time: 16 ms for n = 1000000
Total Sum: 5000000050000000, Time: 4 ms for n = 100000000
Total Sum: 500000000500000000, Time: 33 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg 
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
```

```
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'ThreadedSummation'
Total Sum: 500000500000, Time: 17 ms for n = 1000000
Total Sum: 5000000050000000, Time: 4 ms for n = 100000000
Total Sum: 500000000500000000, Time: 33 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'ThreadedSummation'
Total Sum: 500000500000, Time: 17 ms for n = 1000000
Total Sum: 5000000050000000, Time: 4 ms for n = 100000000
Total Sum: 500000000500000000, Time: 34 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'ThreadedSummation'
Total Sum: 500000500000, Time: 19 ms for n = 1000000
Total Sum: 5000000050000000, Time: 4 ms for n = 100000000
Total Sum: 500000000500000000, Time: 33 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'ThreadedSummation'
Total Sum: 500000500000, Time: 17 ms for n = 1000000
Total Sum: 5000000050000000, Time: 4 ms for n = 100000000
Total Sum: 500000000500000000, Time: 34 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
g:; cd 'g:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2'; &
'C:\Program Files\Java\jdk-23\bin\java.exe' '--enable-preview'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
\Users\saffa\AppData\Roaming\Code\User\workspaceStorage\bea472a
65a985a980eb25f5712a135ca\redhat.java\jdt_ws\exercise_2_a2c4bf9
0\bin' 'ThreadedSummation'
Total Sum: 500000500000, Time: 19 ms for n = 1000000
Total Sum: 5000000050000000, Time: 4 ms for n = 100000000
Total Sum: 500000000500000000, Time: 34 ms for n = 1000000000
PS G:\Other computers\My MacBook Air\School\Shippensburg
University\Fall 2024\CSC523\Exercises\Chapter 7\exercise_2>
```

My expectation is that the serial would be initially better to run while the size is small. However, at some point when the size gets big enough multithreaded would run faster despite the overhead that comes with threading.

Results of Exercise 2:(in much better format)

Serial Version Output:

n	**Sum**	**Time (ms)**
1,000,000	500,000,500,000	2 ms
100,000,000	5,000,000,050,000,000	25 ms
1,000,000,000	500,000,000,500,000,000	486-502 ms

Threaded Version Output:

n	**Total Sum**	**Time (ms)**
1,000,000	500,000,500,000	16-19 ms
100,000,000	5,000,000,050,000,000	4 ms
1,000,000,000	500,000,000,500,000,000	33-34 ms

Analysis of Speed-Up:

- For **n = 1,000,000**, the threaded version took around **16-19 ms**, while the serial version took **2 ms**. In this case, the overhead of thread creation likely caused a slower performance in the threaded version.
- For **n = 100,000,000**, the threaded version was significantly faster at **4 ms**, compared to the serial version's **25 ms**. This shows that for larger values of n, multithreading provides a substantial speed-up.
- For **n = 1,000,000,000**, the threaded version ran in **33-34 ms**, while the serial version took between **486-502 ms**, demonstrating a **huge performance improvement** when using threads.

Conclusion:

- The **serial version** performs better for small values of n due to the overhead involved in managing threads.
- For **large values of** n, the **threaded version** provides a significant speed-up by distributing the computation across multiple threads.