CSCI 2120 - Bonus Assignment: Concurrency

For this assignment you will be solving a computationally difficult problem using concurrency.

In mathematics, a series is the sum of a sequence of terms:

$$\sum_{n=a}^{b} f_n = f_a + f_{a+1} + f_{a+2} + \cdots + f_b$$

Procedure:

Observe the Series.java and SeriesSolver.java classes that have been provided. The Series class represents a series as characterized by an equation for the nth element (using the Java BigDecimal class). The SeriesSolver.java class provides methods for computing the partial sum of a given Series object from one value of **n** (a) to another (b).

A single-threaded implementation

computeSum(Series series, int a, int b);

has been provided for you. This method computes the sum using a *for* loop. Your task is to implement an overload of this method:

computeSum(Series series, int a, int b, int threads);

that performs the same task but using *threads* number of threads. Your code should split the problem into equal-sized sub-problems and then use these answers to generate the overall solution.

1) Implement the computeSum method above.

2) Write a JUnit test for your method comparing the results to the results of the single- threaded implementation. Use the predefined

Series objects provided in the Series class.

BONUS BONUS (+15 points): Using the predefined Series objects in Series.java, benchmark your implementation against the multithreaded version. Provide details about the run time (Using System.currentTimeMillis() for example) of your trials. Write a brief report on your findings. For what range of **n** does the single-threaded method perform faster? What values of *threads* appear to give the best speedup? What type of CPU did you run this on?

Submission

You will add, commit, and push your program to Gitlab. Label your homework folder "Bonus" in your repository. See Moodle for due date.

Grading

This is a bonus assignment. It will count as a 50-point homework grade.