

Lab 8: Java Threads

Instructors: Mack Joyner

Course Wiki: <http://comp322.rice.edu>

Staff Email: comp322-staff@mailman.rice.edu

Goals for today's lab

- Experimentation with Java threads

This lab can be downloaded from the following GitHub repository <https://classroom.github.com/a/g3cSVZ6Z>. Use the git command-line client or IntelliJ to checkout the project into appropriate directories locally.

In today's lab, you need to use NOTS to run performance tests. If you need any guidance on how to submit jobs on NOTS manually, please ask a member of the teaching staff.

1 Conversion to Java threads: Spanning Tree

1. The `SpanningTreeSeq.java` program is an example sequential solution to the spanning tree problem. The `SpanningTreeAtomicHjLib.java` program is a provided parallel solution to the spanning tree problem. This version uses `finish` and `async` constructs along with an `AtomicReference`.
2. Your task is to convert `SpanningTreeAtomicHjLib.java` to a Java program that uses threads instead of `HJlib` tasks. You should modify the provided `SpanningTreeAtomicThreads.java` file, and use Java thread methods instead of `finish/async`. (The `AtomicReference` calls can stay unchanged.) There are TODOs in the file to guide you.
3. You have been provided with tests for your parallel spanning tree implementation in `SpanningTreePerformanceTest`. To complete this portion of the lab, you should submit these performance tests to NOTS by modifying the provided `myjob.slurm` template and submitting manually.

2 Programming Tips and Pitfalls for Java Threads

- Remember to call the `start()` method on any thread that you create. Otherwise, the thread's computation does not get executed.
- Since the `join()` method may potentially throw an `InterruptedException`, you will either need to enclose each call to `join()` within a *try-catch block*, or add a *throws `InterruptedException`* clause to the definition of the method that includes the call to `join()`.

3 Testing on NOTS

Now that we have an implementation of a parallel spanning tree with threads, we will test the performance on the NOTS cluster to measure the actual performance of the implementation without interference on your laptop.

To do so, you should use the provided `myjob.slurm` file. As usual, when using the `myjob.slurm` file please open it to fix any TODO items.

4 Turning in your lab work

For lab 8, you will need to turn in your work by Monday, April 3, 2023 at 3pm, as follows.

1. Show your work to an instructor or TA to get credit for this lab. In particular, the TAs will want to see the output of `testSpanningTreeThreads` running on NOTS through the provided SLURM script.
2. Commit your work to your lab8 folder. Check that all the work for today's lab is in your lab8 directory by opening <https://classroom.github.com/a/g3cSVZ6Z> in your web browser and checking that your changes have appeared.