Lab 7: Recursive Task Parallelism and Cutoff Strategy

Instructors: Zoran Budimlić, Mack Joyner

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

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

Goals for this lab

- Creating parallelism through recursion, using Cutoff Strategy for performance

Lab Projects

Today, we will revisit out simple problem of computing a sum of reciprocal values of an array of doubles, by finish/async and recursion to create parallelism, and a cutoff strategy to optimize actual performance.

The GitHub classroom signup for this lab is located here:

https://classroom.github.com/a/gPbQ6NAS

For instructions on checking out this repository through IntelliJ or through the command-line, please see the Lab 1 handout. The below instructions will assume that you have already checked out the lab7 folder, and that you have imported it as a Maven Project if you are using IntelliJ.

As in Lab 1, you will get a "WARNING: javaagent path does not equal Habanero-Java jar path." warning if you don't set up the javaagent path correctly to point to your local installation of the HJLib .jar file. As in Lab 1, resolve this by editing the run configuration and setting the correct -javaagent option in the VM. For Zoran's setup, this VM option looks like this:

```
-javaagent:"/Users/zoran/.m2/repository/com/github/RiceParProgCourse/hjlib-elog-aedancullen/
master-d2108184ec-1/hjlib-elog-aedancullen-master-d2108184ec-1.jar"
```

For your setup, it will look very similar, but with a different path.

Also, just as with Lab 1, don't forget to pick JDK11 as the JDK for the lab.

1 Parallelization of Reciprocal Array Sum using finish/async, Recursion and Cutoff Strategy

We will now work with the same simple parallel array sum program from Lab 4, introduced in the Demonstration Video for Topic 1.1. Edit the `ReciprocalArraySum.java` program provided in your GitHub repository. There are `TODO`s in the `ReciprocalArraySum.java` file guiding you on where to place your edits.

The goal of this exercise is to create an array of `N` random doubles, and compute the sum of their reciprocals in several ways, then compare the benefits and disadvantages of each. In this lab, you will measure the ACTUAL performance of your code. Your task for today is to:

- Implement the reciprocal array sum in parallel using finish/async and recursion. You can ignore the threshold parameter if you choose to start this way.

- Implement the cutoff strategy for your reciprocal array sum. Use the threshold paremeter for this. You may choose to go directly with this implementation and skip the previous item to save time. It may be easier (but more coding) to complete the previous item first.
- Evaluate different values for the cutoff point by implementing the testCutoffStrategy() method
- Find the optimal cutoff point for your machine and your implementation

2 Turning in your lab work

For lab7, you will need to turn in your work before Wednesday, March 9, 2022 at 4:30PM, as follows:

1. Show your results illustrating which cutoff point achieves the best performance on your machine and your implementation to an instructor or TA to get credit for this lab.
2. Commit and push your work to your `lab7` GitHub Classroom repository. The only changes that must be committed are your modifications to `ReciprocalArraySum.java` and `Lab7CorrectnessTest.java`. Check that all the work for today's lab is in your `lab7` directory by opening `https://classroom.github.com/a/gPbQ6NAS` in your web browser and checking that your changes have appeared.