Lab 4: Creating Parallelism using Data-Driven Tasks

Instructor: Mack Joyner

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

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

Goals for this lab

- Creating parallelism using Data-Driven Futures and Data-Driven Tasks

Lab Projects

Today, we will again solve the simple problem of computing a sum of reciprocal values of an array of doubles, by using data-driven tasks to create parallelism. We will evaluate the performance in this lab using abstract metrics.

The GitHub classroom signup for this lab is located here:

`https://classroom.github.com/a/sLIUD7Or`

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 lab4 folder, and that you have imported it as a Maven Project if you are using IntelliJ.

1 Parallelization of Reciprocal Array Sum using Data-Driven Futures and Tasks

We will again work with the simple parallel array sum program introduced in the Demonstration Video for Topic 1.1. Edit the `ReciprocalArraySum.java` program provided in your GitHub repository. There is one `TODO` in the `ReciprocalArraySum.java` file guiding you on where to place your edits.

The goal of this exercise is to again create an array of `N` random doubles, and compute the sum of their reciprocals. Performance in this lab will be measured using *abstract metrics* that accumulate *WORK* and *CPL* values based on calls to *doWork(1)*. The 4, 8, and max parallel future cases have already been implemented for you.

You need to re-implement the `parArraySum2Futures` using only data-driven futures, data-driven tasks, finish, and asyncs. You should not change the method signature of `parArraySum2Futures`. Don't forget to to call doWork(1) for additions on double values. Integer additions (for example loop indices or array indexing) and other double precision operations should be ignored. For the default input size, our solution achieved an ideal parallelism of *just under* 2.

2 Turning in your lab work

For lab4, you will need to turn in your work before Monday, Feb 12, 2024 at 3PM, as follows:

1. Show your tests passing locally to an instructor or TA to get credit for this lab.

2. Commit and push your work to your `lab4` GitHub Classroom repository. The only changes that must be committed are your modifications to `ReciprocalArraySum.java`. Check that all the work for today's lab is in your `lab4` directory by opening `https://classroom.github.com/a/sLIUD7Or` in your web browser and checking that your changes have appeared.