Lab 1: Infrastructure Setup, Async-Finish Parallel Programming

Instructor: Vivek Sarkar

Course wiki : `https://wiki.rice.edu/confluence/display/PARPROG/COMP322`

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

NOTE: You have the option of doing the setup and execution for today's lab on your laptop computer or on a lab computer. If you use your laptop, the setup should work smoothly if it runs Mac OS or Linux. We will provide tips for Windows users, but we cannot guarantee that the course infrastructure will work on your Windows laptop. For Windows, you should first download and install a standard full JDK (not just a JRE) from Oracle to maximize the chances of DrHJ working on your system. If you're having problems using a Windows machine for your work, we recommend that you use a lab machine instead. You should have 24-hour access to the lab with your Rice ID card.

Finally, all commands below are CaSe-SeNsItIvE. For example, be sure to use "S13" instead of "s13".

1 Coursera Setup

We will use COMP 322's Coursera site, `https://rice.coursera.org/parallelprog-001`, for quizzes and discussions. For all other information, please go to the course wiki listed above. You will need a Coursera account to access this site.

If you can access the above site, you are done and you can move on to the next section. If not, you need to apply for a Coursera account at `http://coursera.org` and send it to comp322-staff@mailman.rice.edu. This is important because you will need to use your Coursera account to complete a lab quiz today.

2 Subversion Setup

Each of you has a private repository for COMP 322 allocated in a "cloud" hosted by Rice's subversion (svn) server, svn.rice.edu. You can always examine the most recent contents of your svn repository by visiting `https://svn.rice.edu/r/comp322/turnin/S13/`*your-netid*.

It is possible that your svn account is not properly set up as yet. If you are unable to access the above URL, please send email to helpdesk@rice.edu cc'ing comp322-staff@mailman.rice.edu and requesting that they fix your access. After that, you can ignore this section for now (till you get access) and move on to the next section.

The svn repository is empty to begin with, but will be populated with folders for homeworks and labs. We have a strict naming convention for these folders — "`hw_1`", "`hw_2`'', ... for homeworks and "`lab_1`", "`lab_2`", ... for labs. There are two ways in which you can turn in files to your subversion repository for lab and homework submissions:

1. You can use a Rice machine called CLEAR as a gateway to publish your data to svn. Rice IT has provided a script called "turnin" to enable you to submit content from a directory in your CLEAR account to your svn repository. Using this approach involves two stages — transferring your files to CLEAR, and transferring files from CLEAR to your svn repository. Both steps are explained below in detail.
2. (For advanced users) If you are familiar with subversion and have your own svn client on your local machine, you are welcome to use that instead.

You can follow the steps below to submit all your labs and homeworks using turnin on CLEAR. You should do your homework and lab work on a different system from CLEAR; it is important to not tie CLEAR down with long-running HJ computations. The instructions below include steps to copy files in a folder to CLEAR, and then to submit them. They are explained for a folder named *lab_1*, but you can use the same instructions for any homework or lab folder.

NOTE for Windows users: To use the following command-line instructions on Windows, you should install a Unix-like command environment for Windows such as Cygwin.

1. Stage 1: Transfer files from your local machine to CLEAR. This stage can be replaced by using a GUI tool such as WinSCP (www.winscp.net) to drag and drop files to CLEAR, if you prefer.

   (a) Go to the folder (in your machine) that contains all the files you need to submit. For now, you can create create a new folder named `lab_1`, and a empty file named *DUMMY.txt* in that folder.

   (b) Zip the directory you want to submit.
   zip -r lab_1.zip lab_1

   (c) Use sftp to copy the zip file to CLEAR.
   sftp <your-netid>@ssh.clear.rice.edu
   <your-password>
   You should see the sftp prompt *'sftp>'* now.
   mkdir comp322
   The above command creates the comp322 directory. It may give an error message if the directory already exists, but that's fine. You can omit this step in the future once you know that the directory exists on CLEAR.
   cd comp322
   put lab_1.zip
   You should see a confirmation that the zip file has been transferred.
   quit

2. Stage 2: Transfer files from CLEAR to the svn repository using turnin. To find out more about the turnin command type the following while logged on to CLEAR: *turnin -help*

   (a) Login to CLEAR
   ssh <your-netid>@ssh.clear.rice.edu
   <your-password>

   (b) Go to the comp322 directory
   cd comp322

   (c) Unzip the file
   unzip lab_1.zip

   (d) Delete the zip file (optional)
   rm lab_1.zip

   (e) Turnin the folder lab_1
   turnin comp322-S13:lab_1
   This should show all your files being added to the subversion. The first time you issue this command you will be asked if you wish to store your password unencrypted, twice.

   (f) Your submission is complete. You will need to repeat these steps at the end of today's lab to submit the work that you've done today.

NOTE: If you have problems with any homework or lab submission during the semester, just email your submission zip file to comp322-staff@mailman.rice.edu before the deadline.

3 DrHJ Setup

3.1 Download

DrHJ is a pedagogic Interactive Development Environment (IDE) for HJ. You can use DrHJ to edit, compile and run HJ programs on whichever machine DrHJ is launched on. In later labs, you will be exposed to command-line interfaces to compile and run HJ programs on different parallel machines without DrHJ.

- Download the jar file for DrHJ from
  `http://www.cs.rice.edu/~vsarkar/downloads/hj/drhj.jar`
- A link to the above jar file can also be obtained by following these links from the course web page: "HJ Info" → "HJ Download and Setup", and then searching for "Download the jar file corresponding to DrJava-HJ"

3.2 Testing

Here are the instructions to compile and run HJ programs using the DrHJ IDE.

- If you're using your laptop, please check that you have version 1.6.x of JDK installed or higher. You can type "java -version" on the command line to check the version.
- Open the DrHJ IDE
  java -jar drhj.jar
- Now you should have the DrHJ IDE running. Check that the compiler options includes "HJ 1.3.1-31680" (the version number may be different in the future). If it does not, then there's a problem and there's no point testing any further. Please contact a lab TA to discuss your problem.
- Download the *HelloWorldError.hj* program from the Code Examples link for Lab 1 in the course web page into a directory called `lab_1`.
- Open an HJ program.
  Click on the open button in the top panel
  Navigate to the folder containing HelloWorldError.hj
  Select HelloWorldError.hj and click open
- Compile the HJ program
  Click on the Compile button in the top panel
- Look at the error messages in both the 'Compiler Output' tab and the 'Console' tab in the bottom panel. Error messages appear in red in the 'Console' tab. You should see an error message as follows: `HelloWorldError.hj:6: Could not find field or local variable "ss".`
- Fix the error by replacing "ss" by "s" and re-compiling. You should now see the message, "`Compilation completed`", in the 'Compiler Output' tab.
- Click on the "Run" button to run the program. You can also go to the 'Interactions' tab in the bottom panel and run the program by typing the following.
  run HelloWorldError
  The `run` command is useful for providing command-line arguments to your program.

3.3 Issues with Windows machines

There are known issues with running DrHJ on some Windows machines. While DrHJ works properly on some Windows computers, the following errors may be encountered on others:

- The HJ compiler is not available as an option when selecting compilers.
- The HJ compiler is unable to locate standard classes such as java.util.*.

If your Windows machine exhibits any of the above problems, then we suggest that you run DrHj on a lab machine, and Xming on your Windows machine to work with the DrHJ display. Instructions on installing Xming can be found at `https://docs.rice.edu/confluence/display/ITTUT/SSH+with+X11+forwarding+on+Windows`.

4 ReciprocalArraySum Program

We will now work with the simple two-way parallel array sum program introduced in Monday's lecture, with a slight modification — this program computes the sum of reciprocals of elements in an array of doubles rather than the direct sum.

You will need to report your work on this problem in a simple quiz using the Coursera site. Follow the link for the Lab 1 quiz at `https://rice.coursera.org/parallelprog-001/quiz/start?quiz_id=17`. It will help you to see the quiz questions ahead of time as you work on this assignment.

- Download the ReciprocalArraySum.hj program from the Code Examples link for Lab 1 in the course web page into the `lab_1` directory.
- The goal of this exercise is to create an array of n random int's, and compute the sum of their reciprocals in two ways:
  1. Sequentially in method seqArraySum()
  2. In parallel using two tasks in (the currently sequential) method parArraySum() with two loops in lines 40 and 42 for lower and upper halves of the array.

  The profitability of the parallelism depends on the size of the array and the overhead of async creation. Your assignment is to use two-way parallelism in method parArraySum() to obtain a smaller execution time than seqArraySum()
- Compile the program by clicking on the Compile button.
- Run the program by typing the following in the Interactions page:
  run ReciprocalArraySum
  Why do you think the answers are slightly different even though the initial version of parArraySum() is sequential?
- Edit the current version to add two-way parallelism to method parArraySum().
- Experiment with different sizes (specified as an integer N) *e.g.,* 10^3, 10^4, 10^5, 10^6, 10^7:
  run ReciprocalArraySum N
  NOTE: You may get an OutOfMemoryError when experimenting with large values of N. You will learn how to address that in a future lab.
- What speedup (ratio of sequential to parallel time) do you see for different values of N? Enter the speedups in a file named `lab_1_written.txt` in the `lab_1` directory.

5 Turning in your lab work

For each lab, you will need to turn in your work before leaving, as follows. This is in addition to submitting the quiz on Coursera:

1. Check that all the work for today's lab is in the `lab_1` directory. If not, make a copy of any missing files/folders there. It's fine if you include more rather than fewer files — don't worry about cleaning up intermediate/temporary files.
2. Before you leave, create a zip file of your work by changing to the parent directory for `lab_1/` and issuing the following command, "`zip -r lab_1.zip lab_1`".
3. Use the turn-in script to submit the contents of the `lab_1.zip` file as a new `lab_1` directory in your turnin directory as explained in Section 2.