Lab 7: Atomic Variables

Instructor: Vivek Sarkar
Graduate TAs: Sanjay Chatterjee, Raghavan Raman

For this lab you shall use the SUG@R cluster to run your programs. More information can be found at `http://www.rcsg.rice.edu/sugar`.

1 Setup on SUGAR

Skip this step if you already have HJ setup on SUGAR.

Do the following steps to setup HJ on SUG@R. Note that you will be pointing to a shared HJ folder which is in /projects. So, you don't need to have a copy of HJ in your home directory on SUG@R.

1. Download the setup file
   `https://svn.rice.edu/r/comp322/course/downloads/hjsetup.txt`
2. Copy this setup file to your sugar account
   scp hjsetup.txt <your-netid>@sugar.rcsg.rice.edu:
3. Login to SUGAR
   ssh <your-netid>@sugar.rcsg.rice.edu
4. Once you are logged in, run the following command:
   source hjsetup.txt
5. Verify that both java and hj are now in your path:
   which java
   which hj

2 Example - N-Queens

1. Download the following 2 programs.
   `https://svn.rice.edu/r/comp322/course/lab7/nqueens_isolated.hj`
   `https://svn.rice.edu/r/comp322/course/lab7/nqueens_atomicInt.hj`
2. Compile both the program for the work-sharing scheduler.
   hjc nqueens_isolated.hj
   hjc nqueens _atomicInt.hj
3. **Do not run your program in your login node!**

3 Running your programs on SUGAR

- Using interactive queue
  1. Request for a interactive session on a compute node
     E.g: qsub -I -N NQUEENS -q interactive -V -l nodes=1:ppn=2,pmem=4000m,walltime=00:30:00
  2. Once the request is granted, you can run HJ programs in it.
     hj nqueens_isolated
     hj nqueens_atomicInt

- You can also run programs on SUGAR by submitting jobs using PBS script

  1. Download the PBS job submission script
     `https://svn.rice.edu/r/comp322/course/lab7/atomic.pbs`
  2. In the script you will find these PBS options :
     - -N : This specifies a job name for you submission
     - -l : With this, you can request you resources from the cluster.
       E.g:
       nodes=1:ppn=8 : We have requested 1 compute node with 8 processors per node.
       Note: Each node has a maximum of 8 processors per node.
       pmem=1000m : Memory requested is 1GB.
       walltime=00:10:00 : Max time for reservation of resources.
       Note: Keeping this much higher than required will delay scheduling of your job
     - -V : This will export your environment to the compute node. This is needed to find paths for hj and java.

     You can change them to what you need when you are experimenting.
  3. At the end of the script you will find the command for running the executable
     hj nqueens_isolated
  4. Now, run the following command to submit the job.
     qsub atomic.pbs
  5. The system will give you back a job id.
     E.g. *3685751.sugarman.rcsg.rice.edu*
     Note: Here 3685751 is the job id.
  6. You can verify the status of your job
     showq | grep <jobid>
  7. Once the job has been completed, you will two files in your directory:
     <jobname>.e<jobid> : This is the error file. E.g: 1D-Iterative-Avg.e3685751
     <jobname>.o<jobid> : This is the output file. E.g: 1D-Iterative-Avg.o3685751
  8. Check the contents of both to see you output or errors.

You can find more information on job submission at: `http://rcsg.rice.edu/sugar/faq/intro.html#PBS`

4 Do it yourself!

You are expected to compare performances for the following two programs.

1. Isolated vs. AtomicReference
   `https://svn.rice.edu/r/comp322/course/lab7/spanning_tree_isolated.hj`
   `https://svn.rice.edu/r/comp322/course/lab7/spanning_tree_atomic.hj`
2. Isolated vs. AtomicIntegerArray

5 Running locally

If you find unexpected delays in seeing results back from SUGAR, you can try to run your programs locally if your local machine has multiple cores.
cat /proc/cpuinfo
The compiler and runtime options should remain the same.