

Spring 2007 – CS 633/733 Grid Computing  
Homework 3

*Due: March 19, 2007 1:00 pm. Individual work only. 200 points.*

Write a program in Java using the Java Commodity (CoG) Toolkit that will perform the following operations:

- a) Read the name of a properties file as a command-line argument (contents and format of the properties file is shown below)
- b) Read the contents of the properties file
- c) Prompt the user to enter the MyProxy passphrase
- d) Retrieve user credentials from the MyProxy server
- e) Copy the input file(s) to the remote host (input file(s) and remote hostname are specified in the properties file)
- f) Submit the job to the host
- g) Wait for the job to complete (the listener should print job status while waiting)
- h) Copy the output file(s) to the specified host(s) (output file(s) and hostname(s) are specified in the properties file)
- i) Destroy the user credentials and exit

Properties file format (similar to RSL parameters):

HOSTNAME = <machine name where the job must run>  
DIRECTORY = <directory from which the job manager will execute the job>  
EXECUTABLE = <executable name with full path>  
ARGUMENTS = <arguments to the executable>  
STDIN = <input file name with full path to be used for stdin>  
STDOUT = <output file name with full path>  
STDERR = <error file name with full path>  
COUNT = <# of processes to be created>  
ENVIRONMENT = <environment variables to be set>  
JOBTYPE = <single or multiple or mpi>  
INFILECOUNT = <# of input files>  
INSRCHOST<sub>n</sub> = <machine name where input file is located>  
INSRCDIR<sub>n</sub> = <input file directory on INSRCHOST>  
INSRCFILE<sub>n</sub> = <input file name on INSRCHOST>  
INDESTDIR<sub>n</sub> = <input file directory on HOSTNAME>  
INDESTFILE<sub>n</sub> = <input file name on HOSTNAME>  
OUTFILECOUNT = <# of output files>  
OUTSRCDIR<sub>n</sub> = <output file directory on HOSTNAME>  
OUTSRCFILE<sub>n</sub> = <output file name on HOSTNAME>  
OUTDESTHOST<sub>n</sub> = <machine name where output file must be copied>  
OUTDESTDIR<sub>n</sub> = <output file directory on OUTDESTHOST>  
OUTDESTFILE<sub>n</sub> = <output file name on OUTDESTHOST>

Based on the number of input file and output files, for each input and output file you will have one set of entries for (INSRCHOST<sub>n</sub>, INSRCDIR<sub>n</sub>, INSRCFILE<sub>n</sub>, INDESTDIR<sub>n</sub>, DESTFILE<sub>n</sub>) and (OUTSRCDIR<sub>n</sub>, OUTSRCFILE<sub>n</sub>, OUTDESTHOST<sub>n</sub>, OUTDESTDIR<sub>n</sub>, OUTDESTFILE<sub>n</sub>), respectively. Use “localhost” as the hostname for coping files to and from the local client. The

Spring 2007 – CS 633/733 Grid Computing  
Homework 3

reason for providing these arguments is to copy the required input file from a different hostname and then copy the output files produced back to another destination host. Not all of the above entries are required; if one of the entries is missing then do not include that entry in the RSL string. Here is a sample properties file (this is just an example):

```
HOSTNAME = everest.cis.uab.edu
DIRECTORY = /home/puri/cs633/homework3
EXECUTABLE = /home/puri/cs633/homework3/psum
ARGUMENTS = 100000
STDOUT = /home/puri/psum-out
STDERR = /home/puri/psum-err
COUNT = 4
JOBTYPE = mpi
INFILECOUNT = 2
INSRCHOST1 = olympus.cis.uab.edu
INSRCDIR1 = /data/usr/puri/cs633/input
INSRCFILE1 = distrib.12252005
INDESTDIR1 = /home/puri/cs633/homework3
INDESTFILE1 = distrib
INSRCHOST2 = localhost
INSRCDIR2 = /home/puri
INSRCFILE2 = params
INDESTDIR2 = /home/puri/cs633/homework3/
INDESTFILE2 = params
OUTFILECOUNT = 2
OUTSRCDIR1 = /home/puri/cs633/homework3
OUTSRCFILE1 = psum.out
OUTDESTHOST1 = olympus.cis.uab.edu
OUTDESTDIR1 = /data/usr/puri/cs633/output
OUTDESTFILE1 = psum.out
OUTSRCDIR2 = /home/puri
OUTSRCFILE2 = psum-out
OUTDESTHOST2 = localhost
OUTDESTDIR2 = /home/puri/psum-out
OUTDESTFILE2 = psum-out
```

**Guidelines:**

1. Compile and run the example program CogExample.java (you can access this from any Vulcan machine at: /mz/mb/puri/cs633/CogExample.java).
2. Review the Java Cog APIs at: <http://www.cogkit.org/release/current/api/jglobus/index.html> and understand the example program.
3. Use Java *properties* class for reading the properties file (use load method) and obtaining the different (name,value) pairs (use getProperty method).
4. You MUST write separate classes for handling the properties, file transfer, job submission, and proxy management. The driver program must invoke these different classes.

Spring 2007 – CS 633/733 Grid Computing  
Homework 3

5. 10% of the grade will be assigned for modularity of your programs. You will be reusing these classes in the next homework.
6. Submit the jobs to jobmanager-sge on the specified host (do not use the default jobmanager).
7. You must test your program with additional input files not just the example shown above.

**Submission:**

Use the “script” command to capture the interactions. Start the session by typing “script”. The inputs and outputs are sent to a file called “typescript” in your current directory. Perform the various operations described in steps (a) – (i) and then type “exit” to end the session (do this after you have debugged your program and ready for the final submission). Create a tar file with all source files, any scripts, typescript file, and instructions to compile and run your programs and upload the tar file to WebCT. Do not make any changes to the scripts on Vulcan after the submission, I will check the timestamps when you demonstrate your homework.