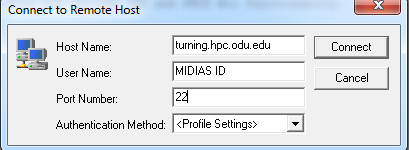
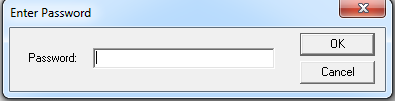
**Implementing Portfolio Optimization using CULA**

Step1. Launching session.

* Open SSH session and connect to turing.hpc.odu.edu
* Give your MIDAS ID as user name and password





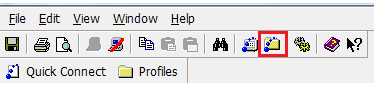
Step2. Create a new directory

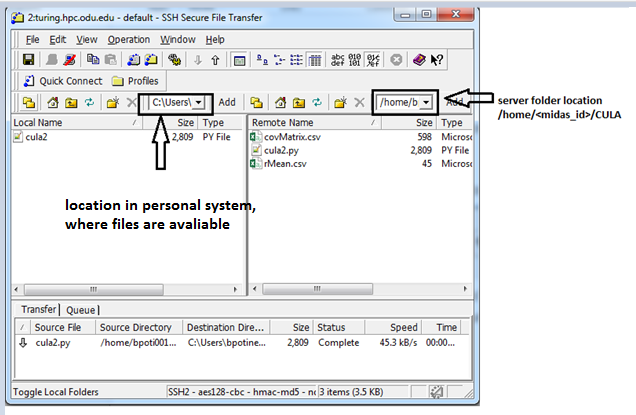
* Create a new directory to import data and code file.

**mkdir CULA**

Setp3. Copy code and data files.

* Select a new file transfer window from the toolbar



Copy data files “covMatrix.txt”, “rMean.txt” and “cula.txt” from resource page under GPU to new directory in turing.hpc.odu.edu server 

* Convert the data files covMatrix.txt and rMean.txt to csv files

**mv rMean.txt rMean.csv**

**mv covMatrix.txt covMatrix.csv**

* Convert code file “cula.txt” to python

**mv cula.txt cula.py**

**Step3.GPU login**

* Use the “gpu-login” script (invokes SGE to allocate an interactive session on a GPU node):

**gpu-login**

step4. Check environment.

* On connecting to appro-006 check for environment variable **$LD\_LIBRARY\_PATH**

*[bpoti001@appro-006 ~/cula]$ echo $LD\_LIBRARY\_PATH*

*/usr/local/cula/lib64:/usr/local/cula/lib:/usr/lib64:/cm/shared/apps/lapack/gcc/64/3.5.0:/cm/shared/apps/gcc/4.9.0/lib:/cm/shared/apps/gcc/4.9.0/lib64:/cm/shared/apps/scipy/0.14.0/lib/:/cm/shared/apps/numpy/1.9.1/lib/:/cm/shared/apps/python/2.7.6/lib/python2.7:/cm/shared/apps/python/2.7.6/lib/:/cm/shared/apps/sge/2011.11p1/lib/linux-x64*

* If not seen the above output please run the fallowing commands

**module load /cm/shared/modulefiles/python/2.7.6  
module load /cm/shared/modulefiles/numpy/1.9.1  
module load /cm/shared/modulefiles/scipy/0.14.0  
module load /cm/shared/modulefiles/gcc/4.9.0  
module load /cm/shared/modulefiles/lapack/gcc/64/3.5.0  
module load /cm/shared/modulefiles/openblas/0.2.14  
module load /cm/shared/modulefiles/cula/18**

Step5. Run code

* By using python run the code on the server from directory

**python cula.py**

Step6. Output

* In output, we print the following
  + Data Read
  + Matrix A, as we need GPU to read it we have to send AT(transpose matrix)
  + Transpose of A
  + X matrix before computing (zero matrix)
  + Matrix B
  + O/p from CULA (using LibCULA SGESV)
  + O/p from python (using solve)
  + Working matrix (Matrix used while computing X using LibCULA SGESV)