

Overview of MSU Compute Servers

The DECS Linux-based compute servers are well suited for programs that are too slow to run on typical desktop computers but do not require the power of supercomputers. The purpose of this handout is to describe how to access a compute server remotely from a Windows PC or MAC.

Additional information about the compute servers that may help you decide which one to use is available on the DECS website (<http://www.egr.msu.edu/decs/facilities/compute-servers>).

NOTE: From the engineering network, users may login directly to the desired computer server using the SSH connection. PC users - PuTTY described below. From outside the engineering network, it is necessary to login to gateway server `scully.egr.msu.edu` and then `ssh` from `scully` to a compute server. More details are described below.

Computer Setup to Run Unix/Linux Programs

Quick Start – for either MAC or PC, **skip the use of the GUI when outside the engineering network**. Most tasks can be done quickly without the GUI. The GUI runs VERY SLOWLY when outside the engineering network.

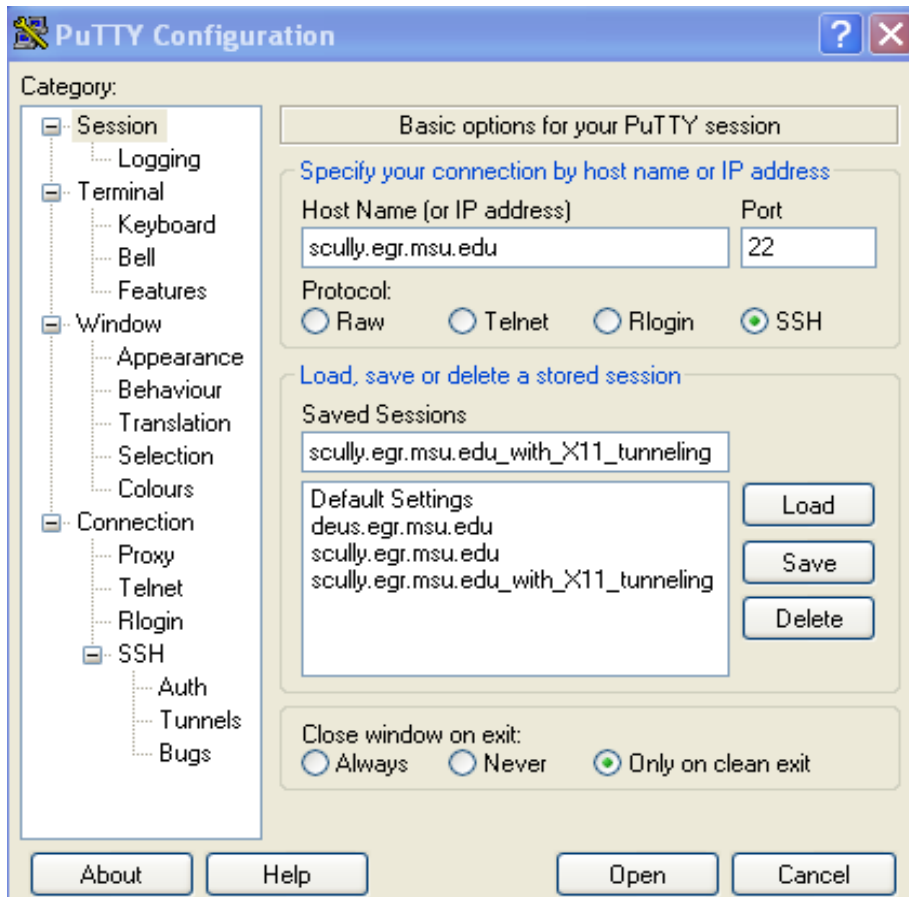
SSH Connections

MACS – the MAC 'XTerm' application can be used to create an ssh connection, no utility is needed.

PCs - PuTTY – needed for SSH connection

Note that PuTTY is already installed on all DECS machines. The communication program Putty is used to connect from a remote computer to the compute server. Putty can be downloaded free from the URL <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. The program is stand-alone and does not need to be installed. Select Putty.exe from the site (you don't need the other programs from the site), store in the desired folder. Create a shortcut in the start menu or your desktop. PuTTY can be used to log into any of the DECS unix or linux machines using the required SSH protocol when on the engineering network. The default settings can be used unless **graphical user interfaces (GUIs)** will be used. If graphical interfaces or plotting will be used, you will also want to 1) Configure PuTTY to use X11 forwarding; 2) Download and install the Xming application (described below).

In PuTTY's Host Name slot, type the name of the machine to which you will login (for `scully`, use `scully.egr.msu.edu`, for `byron`, use `byron.egr.msu.edu`, etc.). Note that customized settings can be saved by entering a convenient name in the Saved Sessions title box and clicking 'Save'.



For use with graphical interfaces, configuring Putty for X11 forwarding for graphical programs: in the Category window, under Connection>SSH>Tunnels, select X11 enabled. Save the configuration.

Configuring for Graphics (Skip if not using Graphics)

MACs – these systems are based on an X-windows system, and no additional software is needed for GUI or plot display.

PCs - INSTALLING the Xming Program

Xming is an X-window emulator. NOTE: Xming is already installed on DECS machines. There are two versions of Xming. The Xming-mesa version is necessary only if the standard Xming version does not work for the application.

This program can be downloaded free from the URL <http://www.straightrunning.com/XmingNotes/>. Download both the open source Xming program and the Xming fonts from the Public Domain Releases table.

Website Releases	Version	State/Notes	Released	MD5 signatures	Size MB
Xming	7.5.0.32	Website Release	27 Jan 2011	MD5 signatures	4.06
Xming-portablePuTTY	7.5.0.32	Website Release	18 Jan 2011	MD5 signatures	1.62
See Donations for how to obtain a Donor Password.					

Public Domain Releases	Version	State/Notes	Released	MD5 signature	Size MB
Xming-fonts	7.5.0.25	Public Domain	24 Aug 2010	587a03fc99bfe0f18bcd4f46d4765eae	30.7
Xming Xming-mesa	6.9.0.31	Public Domain	4 May 2007	4cd12b9bec0ae19b95584650bbaf534a e580debbf6110cfc4d8fcd20beb541c1	2.10 2.50

Once the Xming executable files shown below have been downloaded, run them to install Xming. Administrative rights are necessary to install Xming

Name	Size	Type
Xming-6-9-0-31-setup.exe	2,154 KB	Application
Xming-fonts-7-5-0-25-setup.exe	31,460 KB	Application

Connecting to a Linux server

MACS and PCs - It may save time to run your program on a compute server with a relatively low computational load. To determine which compute servers have low loads, once you are logged into scully or a compute server described below, enter 'computeuse'. The following window will be displayed indicating the relative load on the various compute servers. Logout and reconnect to a machine with a low load if desired. NOTE: A common error is to forget to log into a computer server when using the scully gateway. Matlab will not run on scully. Also scully is not a linux operating system.

HOSTNAME	ARCH	NCPU	LOAD	MEMTOT	MEMUSE	SWAPT0	SWAPUS
brock	1x24-amd64	8	10.66	61.7G	2.8G	9.3G	2.1M
byron	1x24-amd64	8	8.89	61.7G	1.4G	7.5G	0.0
deus	1x24-amd64	8	3.54	61.7G	5.5G	18.6G	116.0K
helper	1x24-amd64	8	7.80	61.7G	4.3G	9.3G	3.8M
monarch	1x24-amd64	8	12.27	61.7G	5.7G	9.3G	3.7M
pinako	1x24-amd64	8	15.23	61.7G	18.7G	18.6G	0.0
rusty	1x24-amd64	8	13.76	61.7G	7.8G	18.6G	0.0
sati	1x24-amd64	8	18.61	61.7G	6.1G	9.3G	0.0
triana	1x24-amd64	8	-	27.5G	-	18.6G	-
winry	1x24-amd64	8	0.00	61.7G	6.1G	18.6G	0.0

MACS -

- 1) Start Applications>Utilities>XTerm.
- 2) Enter 'ssh -Y <username>@<computername>.egr.msu.edu' where <username> is your engineering username and <computername> is the machine where you want to connect.
- 3) If you are within the engineering network, connect to any of the compute servers directly. If you are outside the engineering network, connect to scully.egr.msu.edu, and then after login, use 'ssh -Y <computeserver>' to connect to a compute server. Omit the '-Y' if not using GUI or plots. You will need to enter login information again.

PCs -

- 1) If you will be running a GUI or displaying plots, first start Xming. A small X will appear in the toolbar, but otherwise nothing else will 'appear' to have changed.
- 2) Start PuTTY. If you will be using a GUI, use a configuration that has X11 forwarding enabled. If you are within the engineering network, connect to any of the compute servers directly. If you are outside the engineering network, connect to scully.egr.msu.edu. (Note that when you type your password the cursor will not move – this is normal). The first time you connect to a given server, you will receive a prompt to accept an RSA security key that will be stored on your computer. If the key ever changes after that, you know that the server has changed or that your connection has been compromised.
- 3) Putty will open with a black window. Login using your DECS user information.
- 4) If you are connection via scully, after login, use 'ssh -Y <computeserver>' to connect to a compute server. Omit the '-Y' if not using GUI or plots. You will need to enter login information again.

Running Matlab

After connecting, there are several ways to run Matlab.

Running without the desktop

When tunneling through scully to one of the compute servers using the MATLAB GUI can be very slow. Or you may have already debugged your code and have no need for the interface and the associated use of connection bandwidth. Move to the folder with your matlab files and enter

If you will be displaying plots:

```
matlab -nodesktop -nosplash
```

If not displaying plots:

```
matlab -nodesktop -nosplash -nojvm
```

The GUI will not appear, but MATLAB will prompt you for commands through the terminal window, as shown below. Code without the desktop GUI will still be able to show graphical output if the `-nojvm` option was not used, and for PCs if Xming and X11 forwarding have been configured. To run 'myfile.m', simply type 'run myfile' at the command window prompt. (The 'run' command will not work if the .m files require inputs or return values just use function syntax: [out1,out2,...] = functionname(arg1, arg2,...)). Note that 'cd' and 'dir' commands can be used to navigate/view folders within matlab. To find the present working directory, 'pwd'.

To view a long command window output, it may help to create a 'diary' log. Use diary('output.txt') to append a log of the command window into 'output.txt' or whatever filename is used. Use 'diary off' to stop logging.

To view or edit a file the 'bang' (!) command can be used to run a linux system command from within Matlab. (These commands also work from outside Matlab without the !.) Some common linux commands:

!nano myscript.m – runs the 'nano' editor to edit the file myscript.m.

!less -N myscript.m – view myscript.m with line numbers. PgUp, PgDn, scroll. To quit, press q.

Note: within less, type 'ng' to jump to line n, e.g. 50g jumps to line 50.

!head myscript.m – view the first few lines of myscript.m

lrm output.txt – remove file ‘output.txt’

```
<38 [worden@brock]:~ >matlab -nodesktop
                                     < M A T L A B (R) >
                                     Copyright 1984-2010 The MathWorks, Inc.
                                     Version 7.10.0.499 (R2010a) 64-bit (glnxa64)
                                     February 5, 2010

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

>> █
```

Debugging with the command line interface

When running debugging small jobs, the command line interface works very well. Use ‘help dbstop’ for a summary of commands. Common commands are:

dbstop in <filename> - will stop at first executable line in <filename> when it is run.

dbstop in <filename> at n – will stop at line ‘n’ when <filename> is run.

dbstop if error – will stop if an error occurs so that you can inspect variables.

Helpful line commands at a breakpoint:

dbstep – advance one line

dbcont – continue.

dbstatus – list of breakpoints.

dbtype <filename> <optional first line>:<optional last line> - display file with line numbers.

who – list variables.

whos – list variables, type, and size.

Note: the bang command works at a breakpoint which may be helpful to run ‘less’ to view a file.

Note: enter a variable name to view the contents.

Use ‘quit’ to exit Matlab.

Running batch job without interactive input

If the job is big, it may be most convenient to avoid running the user interface at all. Suppose we want to run a long job ‘myfile.m’. The file ‘myfile.m’ does not create any graphical output, but returns a lot of results to the workspace.

Use of ‘nohup’ will enable you to log out without killing the job.

Create a unix/linux script file using Notepad++ (This is NOT windows notepad) or a unix/linux editor. In Notepad++ be sure to set ‘Format>Convert to UNIX format’. This script needs to be created only one time:

```
#!/bin/csh -f
# Clear the DISPLAY.
unsetenv DISPLAY # unset DISPLAY for some shells
# Call MATLAB with the appropriate input and output,
# make it immune to hangups and quits using 'nohup',
# and run it in the background.
nohup matlab -nodisplay -nodesktop -nojvm -nosplash -r $1 > $2 &
```

All lines that start with # except the first are comments that can be omitted. The first line is required to specify the script language used. **Be sure to follow whitespace carefully and be sure to put a newline after the last line because the last line will 'hang' without a newline character, just as if you type a command but fail to press the 'enter' key.** Filenames you specify on the command line will be substituted for \$1 and \$2 when the script is executed. Save the file with a convenient name such as 'runmatlab.sh' and set executable permissions using 'chmod 700 runmatlab.sh'. The 'sh' extension is commonly used to name script files.

Run the matlab program using

```
runmatlab.sh myfile.m myoutput.txt
```

where you substitute the .m filename for 'myfile.m' the desired command window output file name for myoutput.txt.

The system will return a 'process id' number. Make a note of the server name and the process id. Command window output will go to 'myoutput.txt' or whatever name you designated. The file myfile.m can contain a command to save the workspace, such as 'save myworkspace.mat' or whatever .mat name you choose. The workspace results can later be restored in a new matlab session by using 'load myworkspace.mat'. The workspace results can then be interactively plotted or examined. There are also various methods to create text-based output files.

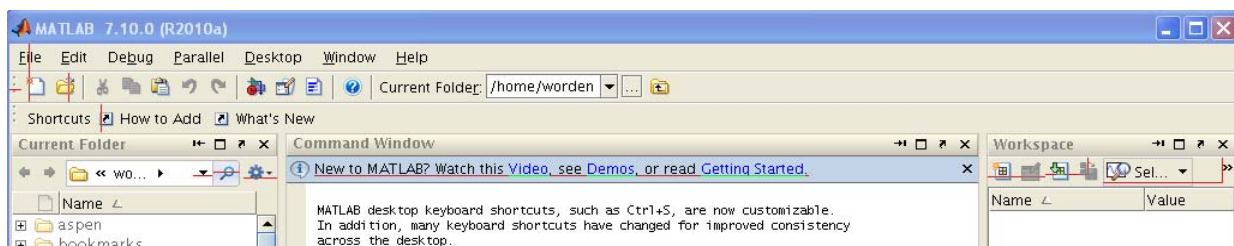
The GUI and command-line methods can be combined by establishing both command-line and GUI connections to the compute servers. The time-intensive computations could be run using the command-line connection and then saving the workspace as a myworkspace.mat file. Then, the GUI connection could be used to load the myworkspace.mat file and plot the results at a later time. You can even create the .mat file on linux and then load it into windows.

To check that a hohup command has finished, you must log into the same server used to start the nohup job. Enter 'ps -umyuser' where myuser is your username. If you recorded the process id or if you submitted only one nohup job, you can verify it the process is finished when it disappears from the list.

Running with the desktop GUI

The first way is to use MATLAB's graphical user interface (GUI) by entering the command 'matlab' at the prompt.

The MATLAB splash screen should show up briefly on the screen, followed by the GUI shown below.



When the GUI window opens, the working directory should be changed to the directory with your files.

When you are done

PCs and MACs- Closing the ssh session

To exit 'nicely' use 'logout' at the ssh prompt to close the session.