Session 1: Accessing MUGrid and Command Line Basics

Craig A. Struble, Ph.D.

July 14, 2010

1 Introduction

The Marquette University Grid (MUGrid) is a collection of dedicated and opportunistic resources spread around Marquette’s campus. The primary resources is pere.marquette.edu, which is a 128 node/1024 core compute cluster funded by the NSF. In addition to the cluster, lab machines in the Department of Mathematics, Statistics, and Computer Science, the College of Engineering, and the Raynor Memorial Libraries contribute idle machine cycles for research computing. Approximately 500 cores are available from idle machines. Figure 1 illustrates the resources available across campus.

Currently, most of these resources run versions of Linux, primarily derived from Red Hat Enterprise Linux. The purpose of this session is to provide instructions on how to access MUGrid systems and use the Linux command line, which is the most common interface to research computing systems worldwide. The skills used to access MUGrid systems are similar to those needed for national grid resources available on TeraGrid and Open Science Grid, to practice on MUGrid will transfer to other resources.

One other note, although Linux is used throughout, the same commands apply to other Unix-like systems such as Solaris, FreeBSD/NetBSD/OpenBSD (with some minor differences), and the Mac OS X command line.

2 Local vs. Remote Systems

Your local system is the computer you are sitting at. Files and programs are (typically) stored on the hard drive in the machine in front of you. You use the local system’s memory, CPU, and peripherals (e.g. keyboard) to get work done.

A remote system is a computer you access through a network or other communications channel. Files and programs are (typically) stored on the remote system and not on the local system you type on. Furthermore, when applications are run on the remote system, the applications use that system’s memory, CPU and peripherals, not the local system’s. Figure 2 illustrates the difference between local and remote systems.

For the remainder of this session, you will be using a lab computer, your desktop computer, or your own laptop as the local system. You will access and use the MUGrid cluster named pere.marquette.edu as the remote system.

3 Accessing pere.marquette.edu

Access to pere.marquette.edu is limited to Marquette University networks. Off campus access is available through Marquette’s VPN and special arrangement with specific labs.

To access pere.marquette.edu users must use a secure shell (SSH) client appropriate for their system. An SSH client provides an encrypted communication channel to access a remote system like pere.marquette.edu. The client provides a basic command line interface (CLI), where you type in textual commands to launch applications and get work done. While there is a significant learning curve to CLIs, a CLI provides flexibility and efficiency in the long term that graphical user interfaces (GUIs) have not
Figure 1: The MUGrid resources.

Figure 2: Local vs. Remote Systems
yet provided. A CLI is the primary means to accessing MUGrid resources as well as national resources on TeraGrid and Open Science Grid.

3.1 Accessing from Windows

On Windows, the free Putty client available at http://www.chiark.greenend.org.uk/~sgtatham/putty/ is easy to install and use. Just download the latest binary from http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe and place it on your desktop. The Putty window is shown in Figure 3. Fill in the “Host Name” as pere.marquette.edu. Once connected, you’ll be asked for your user name and password. Provide your assigned Marquette ID and password.

![Putty configuration window](image)

Figure 3: Putty window after launching.

A commercial option is the official SSH client, which is installed on several Marquette systems. The initial window is show in Figure 4. To login, click the “Quick Connect” button and fill in “Host Name:" as pere.marquette.edu and “User Name:" as your user name. The login window is shown in Figure 5.

3.2 Accessing from Linux and Mac OS X

Linux and Mac OS X systems include SSH clients. On Mac OS X, first start a terminal session by double clicking on the Terminal application in the “/Applications/Utilities” folder. This will bring up a window similar to one shown in Figure 6.
On Linux, several console and terminal applications exist. Refer to your specific distribution documentation to find one.

Once a terminal session has been opened, type in the command

\texttt{\$ ssh username@pere.marquette.edu}

where \textit{username} is your assigned user ID. When prompted for your password, type in your assigned password. You should then be presented with a prompt awaiting commands to run on \texttt{pere.marquette.edu} instead of your local machine.

\textbf{Exercises}

1. Using your assigned username and password, access \texttt{pere.marquette.edu}. When presented with a prompt, type in the command

\texttt{\$ echo Hello World}

You should see the text \textit{Hello World} on the screen and presented with another prompt.
4 Directories

In Linux, files are organized in a hierarchical directory (aka folder) structure with a single root denoted /. A sample of the Linux directory structure on pere.marquette.edu is shown below.

```
/home
  |   
  |   MARQNET
  |     
  |     bobc
  |   
  |   bin
  |   include
  |   lib
  |   local
  |   sbin
```

A path is a sequence of directories separated by path separators, which on Linux is the slash (/) character. The path from the root to the bobc directory in the previous figure is /home/MARQNET/bobc.

Paths can be either absolute, which means they start with the root directory (/) or relative, which means they do not start with the root directory. For example, the relative path to bobc that is relative to /home is MARQNET/bobc. Relative paths are important when combined with the concept of working directories.
4.1 Working Directory

The most important concept for navigating Linux systems like pere.marquette.edu is the working directory. The working directory is your current position in the directory structure and plays important roles when executing commands or editing files. Your working directory is where files will be found by commands unless you specify otherwise.

Display Working Directory

To display your current working directory, use the command

```
$ pwd
```

which stands for “print working directory.”

Relative paths always start from your working directory. It is important to keep this in mind when referring to files and directories using relative paths.

4.2 Home Directories

Every user has a home directory, which is the initial working directory when you log into a Linux system. Your home directory is where you store your documents, data, programs, etc. You can organize your files any way you wish within your home directory and can control who has access to those files using Linux file permissions.
Finding Your Way Home

The primary purpose for introducing home directories now is that you can always “find your way home” by typing the command

```
$ cd
```

which stands for “change directory.” Without anything else on the command line, cd returns you to your home directory.

4.3 Navigating Directories

Navigating directories on a Linux system means to change your working directory. By navigating, you establish a new context for the commands you execute.

To change into a directory, use

```
$ cd dirname
```

where `dirname` is the directory you want to make your new working directory.

To back up one directory in the hierarchy, use

```
$ cd..
```

4.4 Creating Directories

As you generate more files and need to organize them, you will want to create new directories. Use

```
$ mkdir dirname
```

where `dirname` is the name of the directory you want to create.

4.5 Listing Directory Contents

To view the contents of your working directory, type

```
$ ls
```

You will be presented with a list of file and directory names that are stored in your working directory.

To list the contents of a specific directory, type

```
$ ls dirname
```

where `dirname` is the name of the directory you want to view. A similar list of file and directory names in `dirname` will be presented.
Exercises

1. Using the commands above, create the following directory hierarchy rooted at your home directory, which is denoted $\texttt{HOME}$.

```
$HOME
  bin  Documents
     Projects
        First  Second
```

If done correctly, a recursive listing using the command

```
$ ls -R
```

in your home directory produces the output

```
.:         
bin  Documents
./bin:     
./Documents: 
     Projects
./Documents/Projects: 
        First  Second
./Documents/Projects/First:
./Documents/Projects/Second:
```

Write down the command(s) you use to create this directory structure.

2. Navigate so that $\texttt{HOME/Documents/Projects/First}$ is your working directory. Write down the command(s) you use to do so.

3. If $\texttt{HOME/Documents/Projects/First}$ is your working directory, use a relative path and the \texttt{cd} command to make $\texttt{HOME/Documents/Projects/Second}$ your working directory. Write down the command you use.

4. If $\texttt{HOME/Documents/Projects/Second}$ is your working directory, use \texttt{ls} and a relative path to list the directory $\texttt{HOME/bin}$.

5. What are the contents of $\texttt{/group/hpc-bootcamp/Summer2010/Session1}$ on \texttt{pere.marquette.edu}?
5 Editing and Viewing Files

One of the most common tasks performed on remote systems like pere.marquette.edu is to create and edit text files. Scripts, programs, notes, etc. all typically reside in text files that may have special formatting rules.

5.1 Editing Files

A simple to use editor available on pere.marquette.edu and most Linux systems is pico. The editor is started with

```
$ pico
```

From there, you can type in any text that you want. pico will automatically move text to a new line if the line length gets too long. Most of the time this is fine, but when writing scripts and programs later, this is a feature that can cause problems.

You can also directly edit an existing file or create a new file with a specific name by adding a command line argument to pico

```
$ pico filename
```

where filename is the name of the file to create or edit.

5.2 Viewing Files

Frequently, you want to view the contents of a file without editing it. A commonly used command to view files is

```
$ more filename
```

where filename is the name of the file to view. The more command lets you interactively scroll through the file contents one screenful at a time. Use the spacebar to scroll down, b to scroll back (up) and q to quit viewing the file.

You can view the contents of a file all in one go with the command

```
$ cat filename
```

where filename is the name of the file to view. The file contents will scroll quickly in your terminal window. This is useful if you prefer using the scrollbar in your terminal program to navigate the file contents.

Exercises

1. Assuming you have created the directory structure earlier, create a file named README.txt in $	exttt{HOME/Documents/Projects/First}$. Type in your name, the current date, and a brief description of your first MUGrid project. Creating README.txt files is good practice so that you can remember what a project is about 6 months after you last worked on it.

2. View the contents of your README.txt using more and cat commands.

3. View the file in /group/hpc-bootcamp/2010Summer/Session1/Shakespeare.txt using the more and cat commands.
6 Transferring Files

To transfer files between your local system and pere.marquette.edu, a secure copy (SCP) or secure file transfer (SFTP) client is needed. On Windows, WinSCP is a good, free option. SSH also provides a file transfer client in their commercial option. These clients provide graphical, fairly easy to use means to drag files between local and remote systems. The hostname to provide is pere.marquette.edu and username and password are the same as those used when accessing pere.marquette.edu to begin with.

On Linux and OS X, the best approach to transfer files is the scp command. This is similar to the regular copy cp command. Remote files are denoted using a hostname followed by a colon (:).

For example, to copy a file to pere.marquette.edu, use the command

$ scp localfile username@pere.marquette.edu:remotefile

To copy a file from pere.marquette.edu, to the local system reverse the order of arguments.

$ scp username@pere.marquette.edu:remotefile localfile

remotefile is assumed to be in your home directory on pere.marquette.edu, but you can provide both relative and absolute paths to reference other files. You can also recursively copy a directory with the -r option. The command

$ scp -r username@pere.marquette.edu:Documents/Projects/First First

recursively copies the contents of First on pere.marquette.edu to a local directory First in the local working directory.

Exercises

1. Create a README.txt file using a text editor on your local machine. If you’re using Windows, Notepad is a program for creating simple text files. Transfer this file using the SFTP/SCP client available on your system to the remote directory $HOME/Documents/Projects/Second.

2. View the contents of the README.txt file using more or cat on the remote system pere.marquette.edu.

3. Transfer the file /group/hpc-bootcamp/2010Summer/Session1/Shakespeare.txt from the remote system pere.marquette.edu to your local system. Use a local text editor or viewing program (e.g. Notepad) to read the file.

7 Other Useful Commands

Copying, Renaming, and Moving Files

Copy a file

$ cp origfile newfile

Rename a file
$ mv origname newname

Move a bunch of files into a directory

$ mv file1 file2 ... dirname

Note that dirname must already exist to move files into the directory.

Removing Files and Directories
Remove a bunch of files. Beware there's no going back.

$ rm file1 file2 ...

Remove an empty directory.

$ rmdir dirname

Recursively remove a directory and its contents. Beware, this is powerful and dangerous.

$ rm -r dirname

Useful Information
Read the online manual page for a command.

$ man command

See what processes you are running.

$ ps

See the processes you and others are running.

$ ps -efww