

Remote Connection to the Zoo

1 Introduction

The Zoo computers can be used in three different ways:

1. Come to the Zoo and log in directly on one of the Zoo consoles with the new 24" displays.
2. Log in from another computer via `ssh` and use the command line tools.
3. Create a Virtual Network Computing (VNC) remote Zoo desktop on another computer. This provides a full graphical interface to the Zoo environment.

Being there in person is the most satisfying and trouble-free method. It has the added advantage of putting you in the laboratory with others who can often answer your technical questions. Connecting via `ssh` is fast and easy, but you are limited to Zoo applications that have command line interfaces. To make remote use of graphical tools such as the Eclipse¹ integrated development environment, you will want to set up a VNC connection. This document explains how.

2 Why not run Eclipse (or Xcode or Visual C++) locally?

Eclipse is by no means the only or even best IDE for developing C++, nor is the latest version of `g++` installed on the Zoo (version 5.3.0 in `/usr/local/gcc-5.3.0`) the only good C++ compiler. The reason for preferring the tools on the Zoo is that the Zoo is the target environment for the programs you develop in this course. While code developed on another platform *might* compile and run correctly on the Zoo, it also might not. The reasons for such *compatibility problems* are many. The development environment might fail to flag errors in your program which happen to violate the standard and hence are not portable to another system. It might provide optional tools or libraries that are not available on the Zoo. It might provide a different version of the tool which doesn't work quite the same as the version on the Zoo.

The best way to avoid compatibility problems is to use the *same* software that will be used to grade your assignments. That means, develop and test on the Zoo, not elsewhere. VNC makes this possible even when you cannot be physically present in the Zoo.

3 What do I need to run VNC?

You will need two things: an SSH (secure shell) client and a VNC viewer.

¹Please note that there are two versions of eclipse installed on the Zoo. The one you want is "Mars.1" (at `/usr/local/eclipse-4.5.1/eclipse`), *not* the older "Mars" version at `/usr/bin/eclipse` that you get if you just type `eclipse`.

3.1 SSH client

SSH is used both to make the initial contact with the Zoo and to establish an encrypted tunnel between your machine and the Zoo for use by VNC.

Most Linux distributions and Mac OS X already have `ssh` installed. For Windows machines, you will need to install an SSH client such as `putty` that supports tunneling. Putty can be downloaded here: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

3.2 VNC viewer

The VNC viewer on your machine talks to the VNC server on the Zoo, so you will need to install a VNC viewer if one is not already installed. Many Linux distributions furnish a VNC viewer, although it won't necessarily be installed by default. The Zoo machines provide the `TigerVNC` server and viewer. Other VNC viewers such as `RealVNC` can also be used, although they won't necessarily all give equally good performance.

The `TigerVNC` viewer for Windows and Mac can be downloaded here: <https://bintray.com/tigervnc/stable/tigervnc/1.6.0>.

4 Establishing a VNC Connection

Establishing a VNC connection logically involves four steps:

1. Use `ssh` to log into a Zoo node.
2. Start a VNC server on the Zoo node.
3. Create an `ssh` tunnel.
4. Start a VNC viewer on your machine to connect to the Zoo server over the tunnel.

Depending which SSH client and VNC viewer you are using, some of these steps can be combined. For example, most SSH implementations allow you to log in to a remote node and establish a tunnel (using port forwarding) at the same time, thereby combining steps 1 and 3. On the other hand, some VNC viewers will create an SSH tunnel and then connect through it, combining steps 3 and 4.

4.1 Using SSH to connect to a Zoo node

4.1.1 SSH on Linux and Mac OSX

Linux machines and Macs with OSX should have a version of OpenSSH already installed. Open a terminal window, run the following command, and follow the prompts to log in:

```
ssh <username>@node.zoo.cs.yale.edu
```

The special name `node.zoo.cs.yale.edu` selects one of the Zoo nodes automatically for you. *Important: Make a note of which node you are actually connected to in case you need to go back to it later.*

4.1.2 SSH Tunneling with Putty on Windows

The instructions below describe how to use Putty on Windows to both open an SSH session to a Zoo node and also to forward the VNC over the same connection. The first VNC server to be started on a node uses port 5901, the second 5902, and so forth, so these are the ports that need to be forwarded from the Zoo node back to your machine.

1. Connect to a free node in the zoo by using `node.zoo.cs.yale.edu` as the host. You can also substitute a specific node name.
2. To avoid having to perform all of these steps every time you bring up putty, enter this same value in the saved sessions textbox and click “save.”
3. Select the new item from the saved session list and click “load.” You will now be editing the details for your saved session information.
4. Before connecting, go to Connection > SSH > Tunnels on the left menu.
5. You will need to add the ports that you will be using. Usually the ports will be 5901 or 5902, but if there is a lot of activity you may have higher ports. For each port you add, in the destination text field, add `localhost:portnumber`. So, to add port 5901, you should put “`localhost:5901`” in the destination field.
6. Once you have added some ports, go back to the Session option on the left panel.
7. Make sure “only on clean exit” is selected at the bottom before connecting.
8. Click “save.” This step is important since it will save the session information you just edited so that you don’t have to add all the ports again the next time you want to connect. For future connections, you can just select the correct option in the saved sessions list and click “load” before connecting.
9. To connect, click “open.” Once a command prompt pops up, log yourself in.

4.2 Setting up the VNC Server

Regardless how you have connected to the Zoo, here are the instructions for starting the VNC server.

1. Go to the command line interface created by SSH. This will be Putty in Windows, or the command prompt used in the immediately preceding step for Linux/OSX.
2. Run the command `vncserver`. To set the window size, run `vncserver -geometry 800x600` (or whatever your preferred resolution is) instead of just `vncserver`.
3. If this is the first time you have used `vncserver`, you will be asked to set up your server at this point, and will need to supply a password for it. You will be asked for this password when running the VNC Viewer.
4. Once set up, you will get a message like this:

```
New 'X' desktop started on grizzly:1
```

Make note of the node name and the number at the end, which is the X display number. Your server is listening on port $5900+n$, where n is the display number. When using Putty with SSH tunneling, that port must have been added previously to the list of ports to be forwarded to the local node.

If $5900+n$ is higher than any port you added when configuring Putty, you'll have to close Putty, restart it, add the new port(s), and then connect again. Before closing Putty, be sure to shut down the server cleanly as described in Section 5.

4.3 Setting up the VNC viewer

4.3.1 On Windows

If using Windows, open up TigerVNC Viewer and tell it to connect to `localhost:n`. From the example above, this would be `localhost:1`. You should get a window asking for the password you created earlier.

Do not make the mistake of connecting directly to `grizzly:n`. This will also work, but it will open an ordinary (unencrypted) TCP connection to the node “grizzly”, and your connection password will go out in the clear. By connecting to `localhost:n`, you are sending your connection through the SSH-encrypted tunnel previously created.

4.3.2 On Mac

If using Mac OS X, you need to create a tunnel before running TigerVNC. From another Terminal window, if your display number is 1 (as explained above) execute

```
ssh -L 5901:localhost:5901 username@nodename.zoo.cs.yale.edu
```

otherwise change the port to $5900+n$, where n is the display number. Here, *nodename* is the name of the node you connected to in step 4.1.1

Now open up TigerVNC Viewer and tell it to connect to `localhost:n`. From the example above, this would be `localhost:1`. You should get a window asking for the password you created earlier.

Note: the first time you access the remote desktop you will probably get a black screen. If that is the case, please refer to section 6.2 below to learn how to set up the window manager to be used when running the VNC server.

4.3.3 On Linux

With Linux, it is possible to set up port forwarding with the initial SSH connection and then connect the VNC viewer to `localhost:n` as described above for Windows. However, the Linux VNC viewer command supports the `-via` option that creates an SSH tunnel and connects the viewer through it all in one go.

From another window, give the command

```
vncviewer -via nodename.zoo.cs.yale.edu localhost:n
```

Following the example above, this would be

```
vncviewer -via grizzly.zoo.cs.yale.edu localhost:1
```

After entering the password created when running `vncserver`, a remote desktop should show up.

5 Exiting Cleanly

When you are done using the Zoo node, you must exit cleanly and make sure you do not leave behind running `Xvnc` server processes. (The command `vncserver` is used to create and terminate the server; the server itself is called `Xvnc`.) If you forget to kill the server, it will continue to run even after you log out. The next time you or anyone else runs `vncserver`, a new `Xvnc` process is created and assigned the next larger display number.

The safest way to clean up is to issue the command

```
vncserver -kill :n
```

where `:n` is `:1` or `:2` etc. depending on the display number `n` that was shown to you when the server started. You can also use the command

```
killall Xvnc
```

to kill all of the `Xvnc` server process that you own.

6 Using the Remote Desktop

When the remote desktop is opened, it starts a new X session. X is very general and can start any of several different window managers. By default, VNC uses the simple `twm` window manager.

6.1 Using `twm`

When the remote desktop first opens under `twm`, you should see a window with a command prompt showing. You can use this as you would a command prompt on the Zoo. If you type `eclipse`, you may see a strange grid show up and nothing will happen. This will happen any time a window pops up. The grid allows you to pick where you want the new window to appear by clicking somewhere on the screen.

If you accidentally close the command prompt, click and hold anywhere on the remote desktop and select “Xterm.” To move files back and forth between your computer and the one you are remotely connected to, you will still need to use `putty` or another program like `WinSCP`.

Eclipse may occasionally experience some strange problems when being used remotely. Parts of the window may not display correctly (being a black box or blank space), and the program itself may stall. The entire remote desktop may stall as well, especially if your Internet connection is very slow or unreliable (particularly for wireless connections where the signal strength varies a lot over a short period of time). If this happens and you are unable to fix the problem from within the remote desktop window itself, you may still exit cleanly as described below and force the remote desktop to disconnect. To be safe, after running `vncserver -kill :n`, check for any residual processes running before closing the command prompt used to start `vncserver`. You do not want to leave orphaned processes behind when you exit.

6.2 Using `Gnome` or `KDE`

When logging in locally from a Zoo console, most users choose to use either the `Gnome` or the `KDE` desktop. To have the same experience when using VNC, you will need to tell the VNC server what window manager or desktop to run.

When the VNC server starts, it runs the script

```
~/vnc/xstartup
```

The first time you run VNC server, it will create such a script for you if it doesn't already exist. The one it creates will probably start the `twm` window manager described above. For example, here is the script that VNC created for me:

```
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
```

This script loads my `.Xresources` file, sets the background color of the root window, start an `xterm` process, and then passes control to the `twm` window manager.

I chose to replace `~/vnc/xstartup` with

```
#!/bin/sh
xsetroot -solid grey
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
[ -r $HOME/.Xmodmap ] && xmodmap $HOME/.Xmodmap
startkde
killall Xvnc
```

This sets the background color, loads my `.Xresources` and `.Xmodmap` files (to change key bindings) only if those files exist in my home directory, and then it starts up KDE. If the user logs out of KDE, then the script kills all running `Xvnc` servers that the user owns.

7 Automating the Connection from Linux Clients

If you are running a Linux client, you can use the following script to automate the process of establishing a VNC connection to the Zoo. Here is the script `zoovnc` that I use:

```
#!/bin/tcsh -f
#
# Start vnc viewer on zoo node and connect to it

set hostWin='(ssh node.zoo.cs.yale.edu vncserver -geometry 1540x1120) |& \
sed -n "s/^New '[^']*' desktop is //p"'

echo $hostWin

if ( $%hostWin == 0 ) then
    echo "Connection failed"
    exit 1
endif
```

```
set remoteHost='echo $hostWin | awk -F: '{print $1}''  
set remoteWindow='echo $hostWin |awk -F: '{print $2}''  
  
echo "New 'X' desktop started on ${remoteHost}:${remoteWindow}"  
echo "Starting vncviewer on $remoteHost"  
  
vncviewer -via $remoteHost localhost:$remoteWindow  
  
echo "Shutting down Xvnc on host $remoteHost"  
ssh $remoteHost vncserver -kill :$remoteWindow
```

This script first uses `ssh` to connect to the Zoo and start a server. Next, it extracts the host name and window number to the shell variables `remoteHost` and `remoteWindow`, respectively. It then starts the VNC viewer. When the viewer exits, it goes back to the host and kills the server in case it is still running.

8 Acknowledgment

We gratefully acknowledge Dr. Donya Quick who wrote an earlier version of this document and to Luciano Dybala for adding the section on connecting from a Mac using TigerVNC.