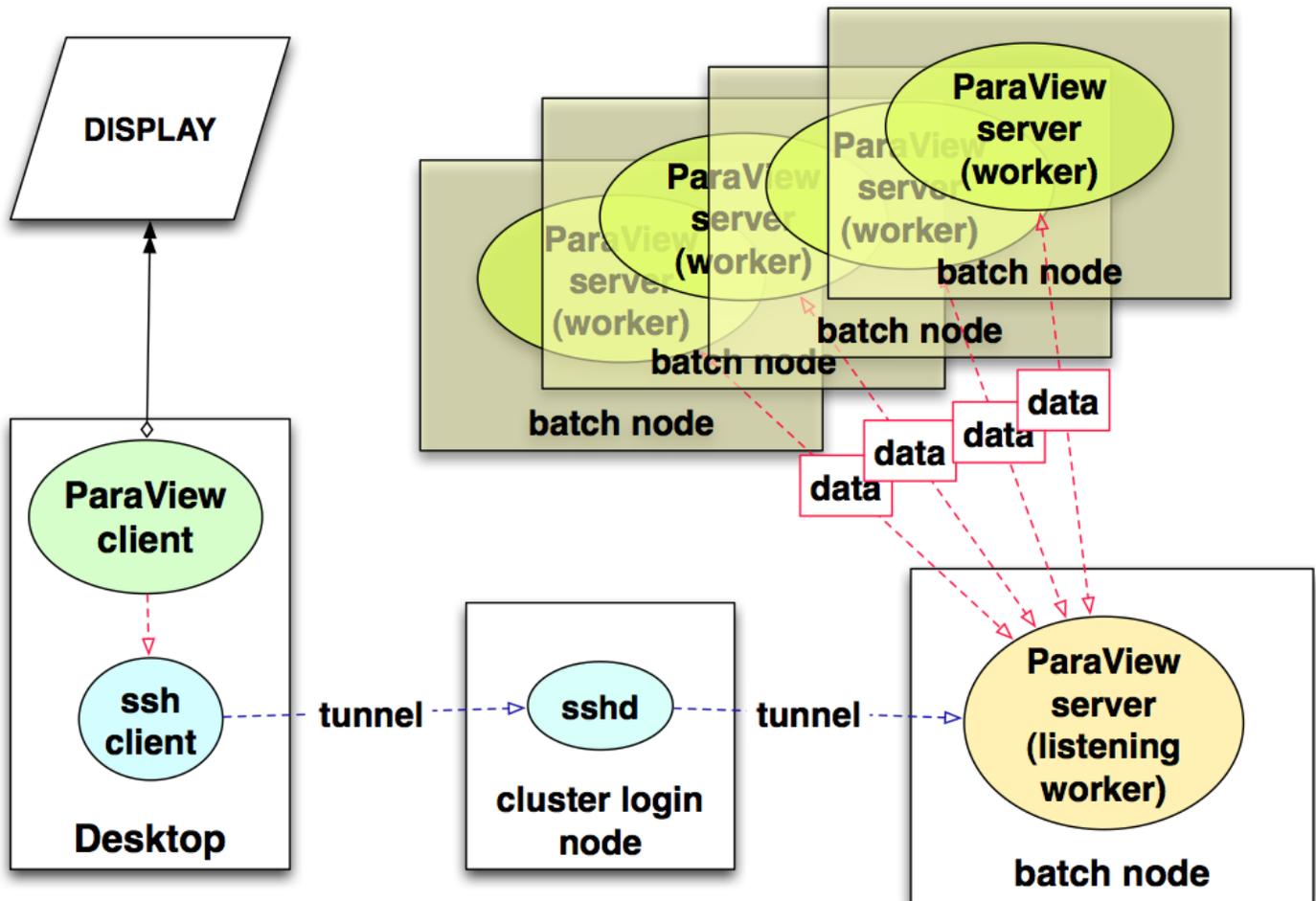


Running ParaView in Client-Server mode

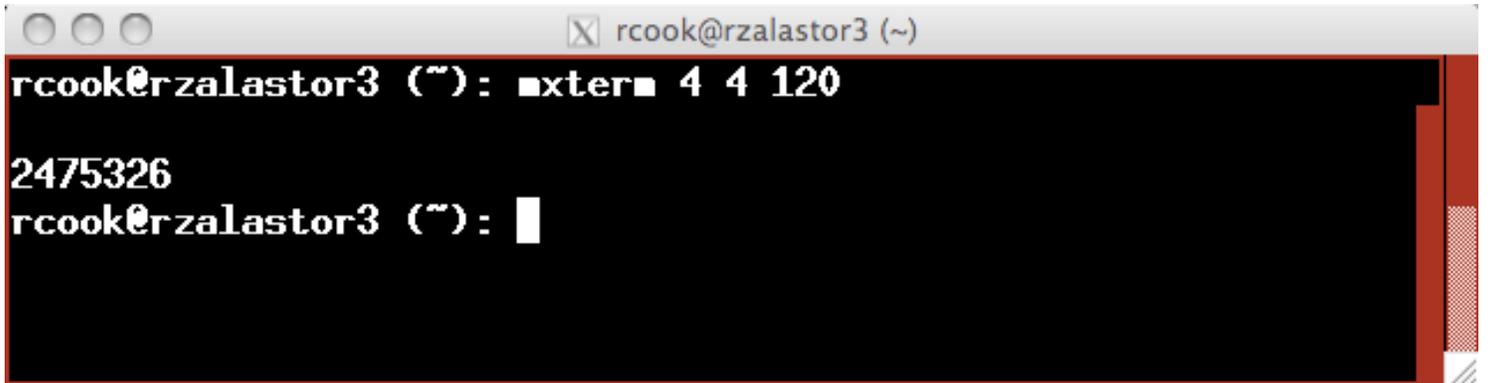
This lesson shows you how to run ParaView with the client running on your desktop and the pvservers executing in parallel on the LC clusters. This tutorial assumes you are using a Macintosh or Linux desktop machine. If you are using Windows, it will still work, but you will need to use something like PuTTY to set up the SSH tunnel. Please contact the LC Computing Hotline at 2-4531 for assistance.

Overview of ParaView components for parallel usage at LLNL



Paraview at LLNL in parallel is shown pictorially in this figure. The client runs on your desktop and uses your graphics card to display images quickly on your computer. The heavy lifting is done on batch nodes on some cluster. Because of the firewalls at LLNL, you will have to establish an ssh tunnel to carry the data from the "listening server" to the client on your desktop. The instructions below walk you through creating this setup.

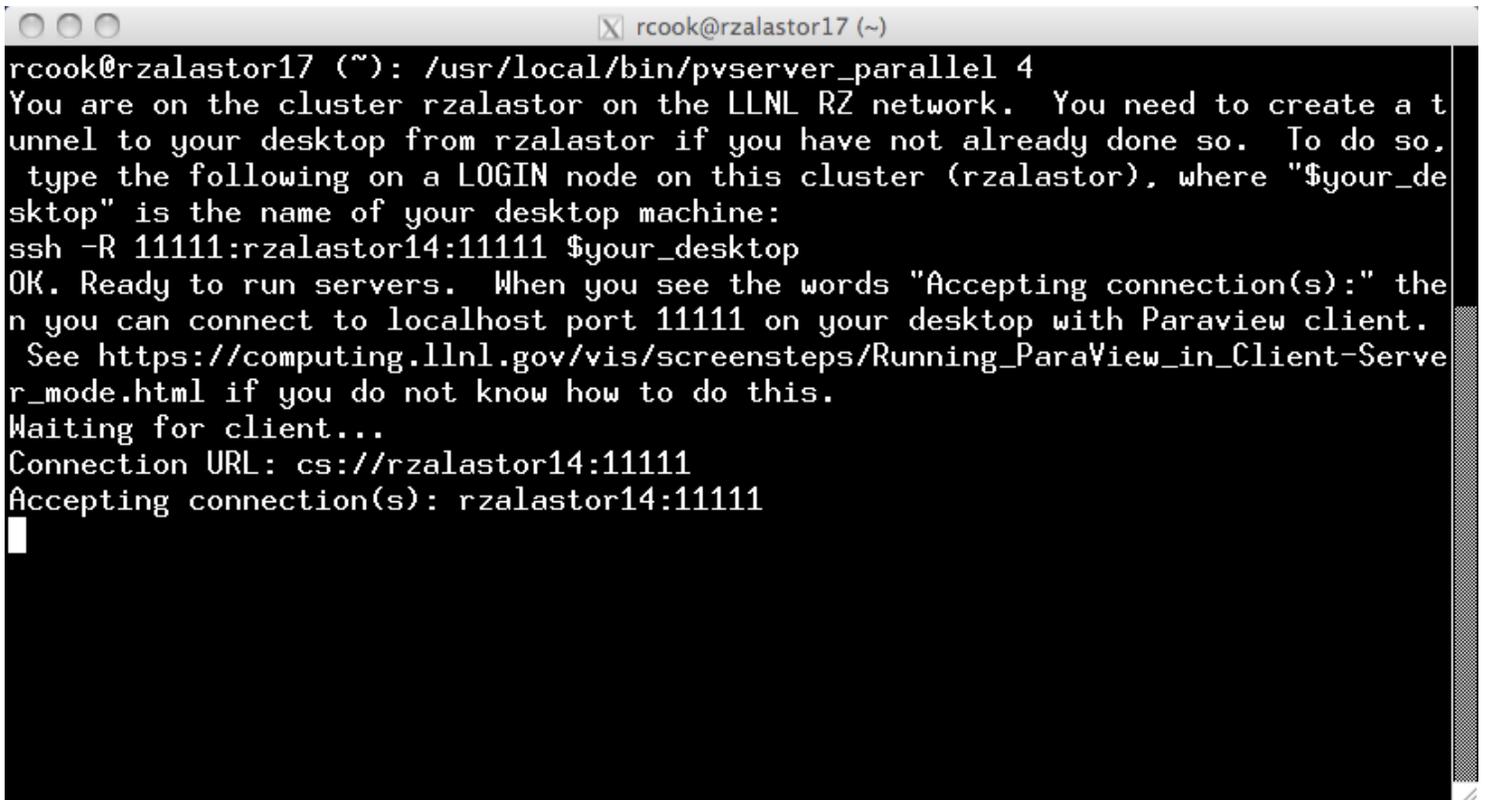
Start a batch job



```
rcook@rzalastor3 (~): mxterm 4 4 120
2475326
rcook@rzalastor3 (~):
```

Start a batch job to reserve some nodes. The mxterm command at LLNL is handy for this purpose. The above commands reserves 4 nodes for 120 minutes.

Start the pvservers on the batch nodes and create an ssh tunnel if needed.



```
rcook@rzalastor17 (~): /usr/local/bin/pvserver_parallel 4
You are on the cluster rzalastor on the LLNL RZ network. You need to create a tunnel to your desktop from rzalastor if you have not already done so. To do so, type the following on a LOGIN node on this cluster (rzalastor), where "$your_desktop" is the name of your desktop machine:
ssh -R 11111:rzalastor14:11111 $your_desktop
OK. Ready to run servers. When you see the words "Accepting connection(s):" then you can connect to localhost port 11111 on your desktop with Paraview client. See https://computing.llnl.gov/vis/screensteps/Running_Paraview_in_Client-Server_mode.html if you do not know how to do this.
Waiting for client...
Connection URL: cs://rzalastor14:11111
Accepting connection(s): rzalastor14:11111
```

When the nodes are reserved, an xterm window will appear, and you will be logged into the **last** node in the batch job. In the above image, it's rzalastor17. Closing this window will terminate your batch job. In this window, type the following command (you can copy and paste). **Make sure the version number matches the version of paraview you are using on your desktop.** Here we are launching version 4.2.0:

```
/usr/local/bin/pvserver_parallel-4.2.0 4
```

This runs four pvserver instances in parallel. You can pass any option to pvserver_parallel which pvserver would take.

Note that it gives you instructions for creating an SSH tunnel **if you have not already done so**. The purpose of the SSH tunnel is to connect your desktop to the server node. Follow the instructions exactly as shown.

The instructions are different depending on whether you are on a CZ machine or an RZ machine. Please pay close attention.

In the example, please note that the tunnel is to rzalastor14, not rzalastor17 as you might think. Paraview listens on the first batch node, but mxterm puts you on the last batch node. Not my fault!

Windows only: setting up an SSH tunnel with PuTTY

The image shows a terminal window at the top with the following text: "create a tunnel from your desktop to this cluster on the CZ, if you have already done so. To do so, type the following on your desktop machine: ssh -L 11111 surface24:11111 surface OK. Ready to run servers. When you see the words 'Accepting connection you can connect to local host port 11111 on your desktop with Paraview. See https://computing.llnl.gov/vis/screensteps/Running_ParaView_in_Cl".

Below the terminal is the PuTTY Configuration dialog box. The "SSH" category is selected in the left sidebar. The "Options controlling SSH port forwarding" section is expanded. The "Port forwarding" section has two checkboxes: "Local ports accept connections from other hosts" and "Remote ports do the same (SSH-2 only)", both of which are unchecked. The "Forwarded ports:" section is empty. The "Add new forwarded port:" section has a "Source port" field containing "8080" and a "Destination" field containing "www.ccs.neu.edu:80". The "Add" button is highlighted. The "Local" radio button is selected under the "Destination" options.

Four red arrows with numbers 1, 2, 3, and 4 point to specific elements: Arrow 1 points to the "SSH" category in the sidebar; Arrow 2 points to the "Source port" field; Arrow 3 points to the "Destination" field; Arrow 4 points to the "Add" button.

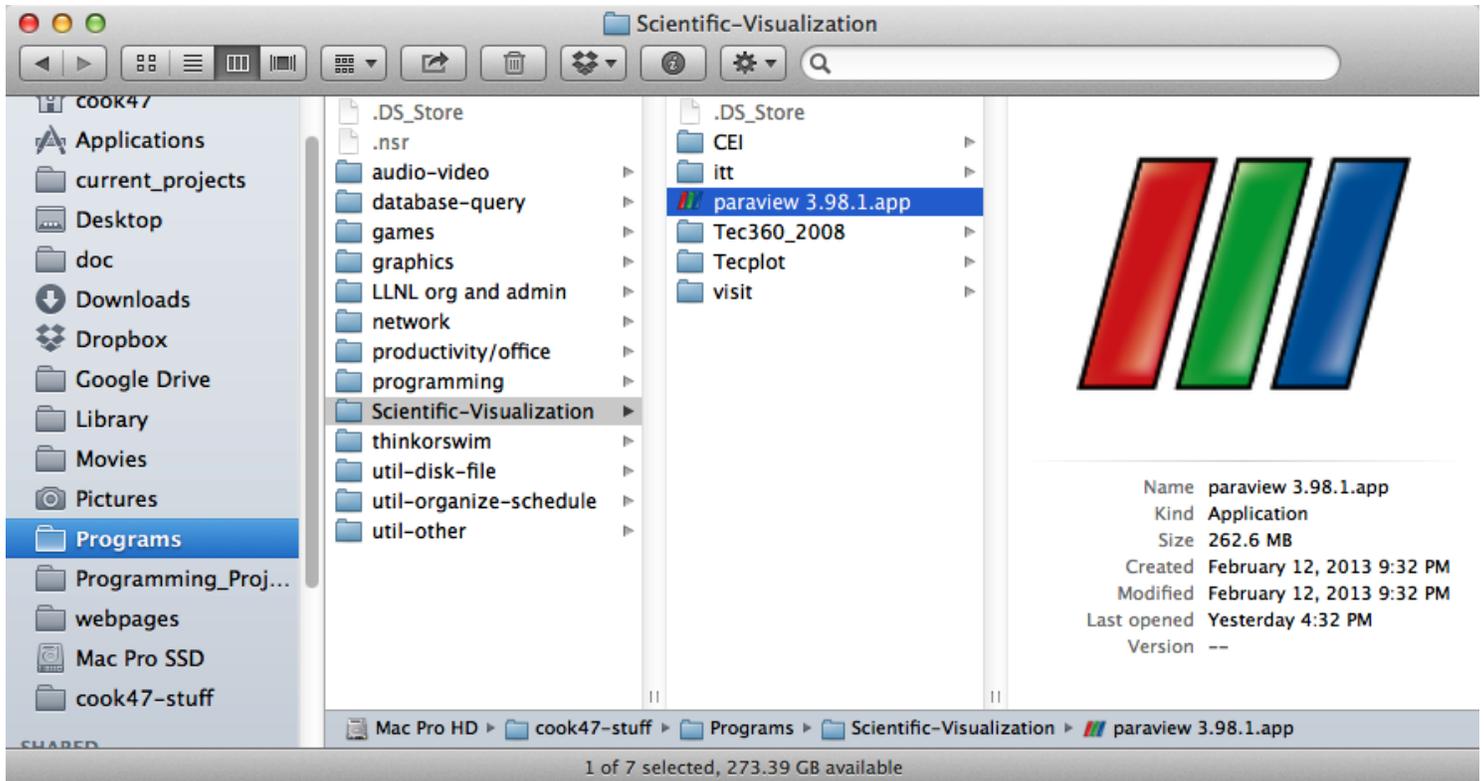
To the right of the dialog, the text "Click 'Add', then connect to 'surface' in next dialog" is written in red.

On Windows, you can set up a tunnel with PuTTY. This picture is intended to show how to map the instructions from pvserver_parallel to PuTTY's "tunnel" dialog. Of course, you must refer to the actual output of pvserver_parallel from the previous step to get actual values for your batch job.

1. The -L means "Local". If you see -R instead, try the "Remote" button.
2. The number "11111" before the first colon is the "source port"
3. Type the host:port, e.g. "surface24:11111", into "Destination".
4. When done, click "Add", then connect to "surface" for this example.

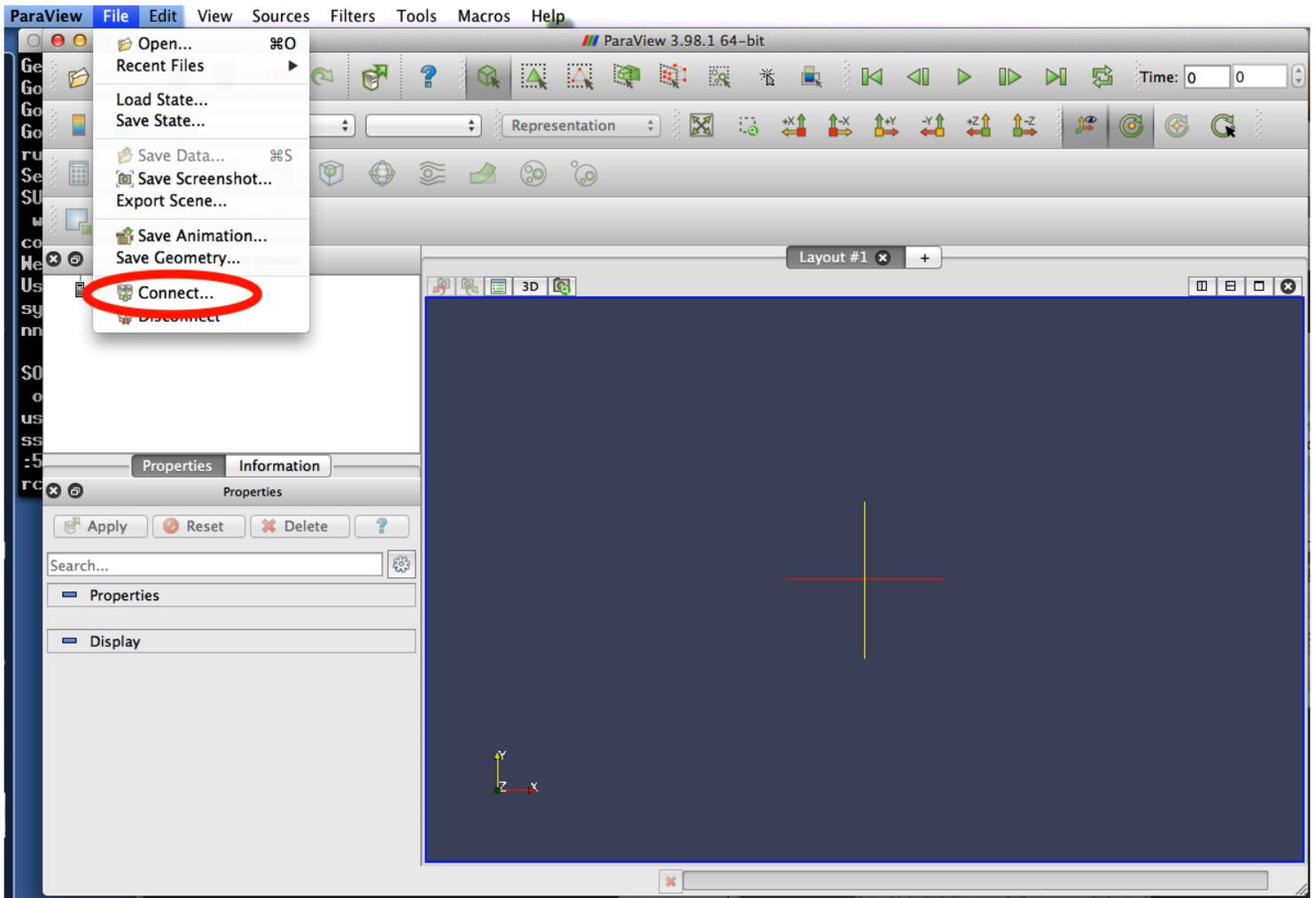
Again, this is only for Windows. On Mac or Linux, just follow the instructions from the previous step exactly as given. If you are confused, call the Hotline and they will call me. :-)

Launch ParaView on your desktop



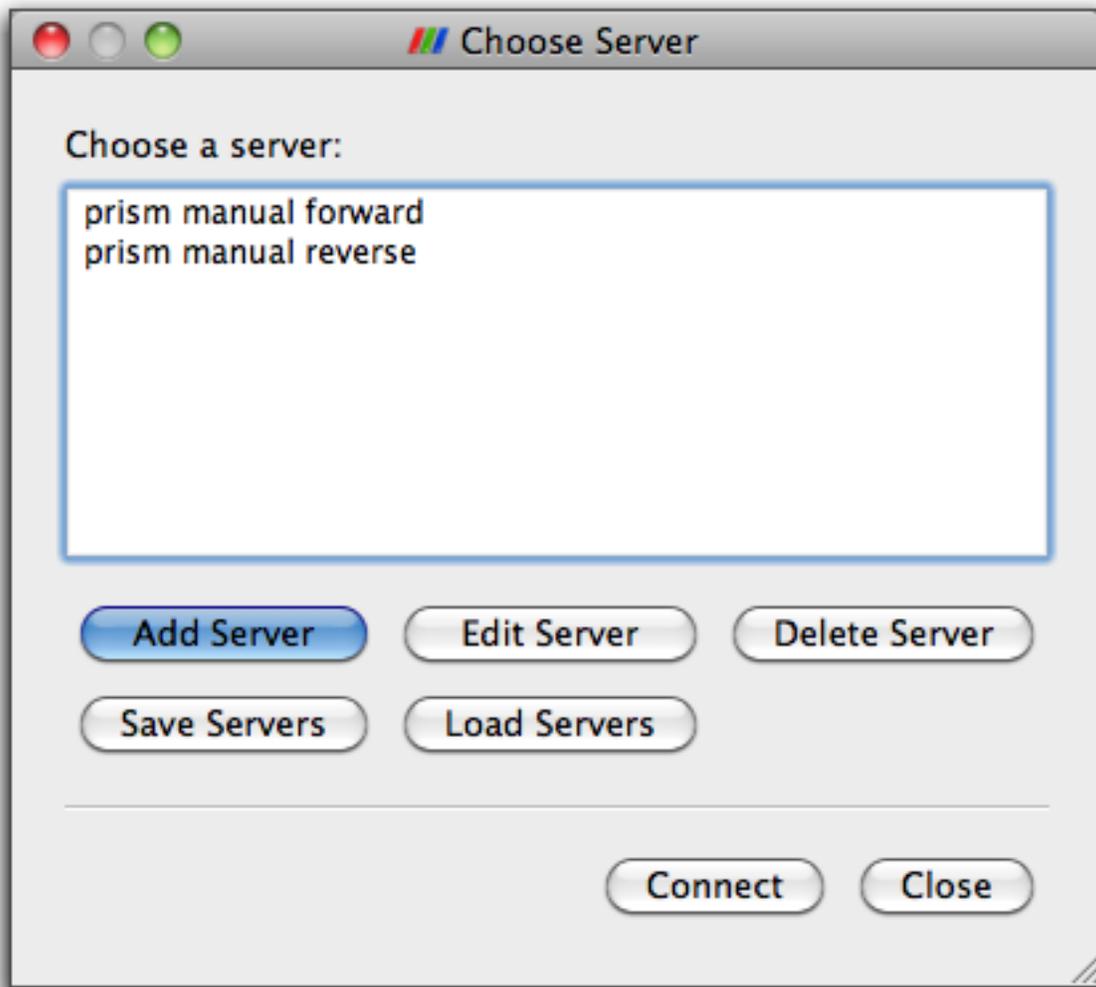
Launch the ParaView client you have installed on your desktop machine. I've tucked mine away a few directories deep. Make sure the version matches the version installed on the LC clusters. Currently that's 3.98.1. Type `/usr/local/bin/pvserver --version` on any cluster to find out what version is current.

File --> Connect



Choose "Connect" in the File menu.

Add Server (if not already added previously)

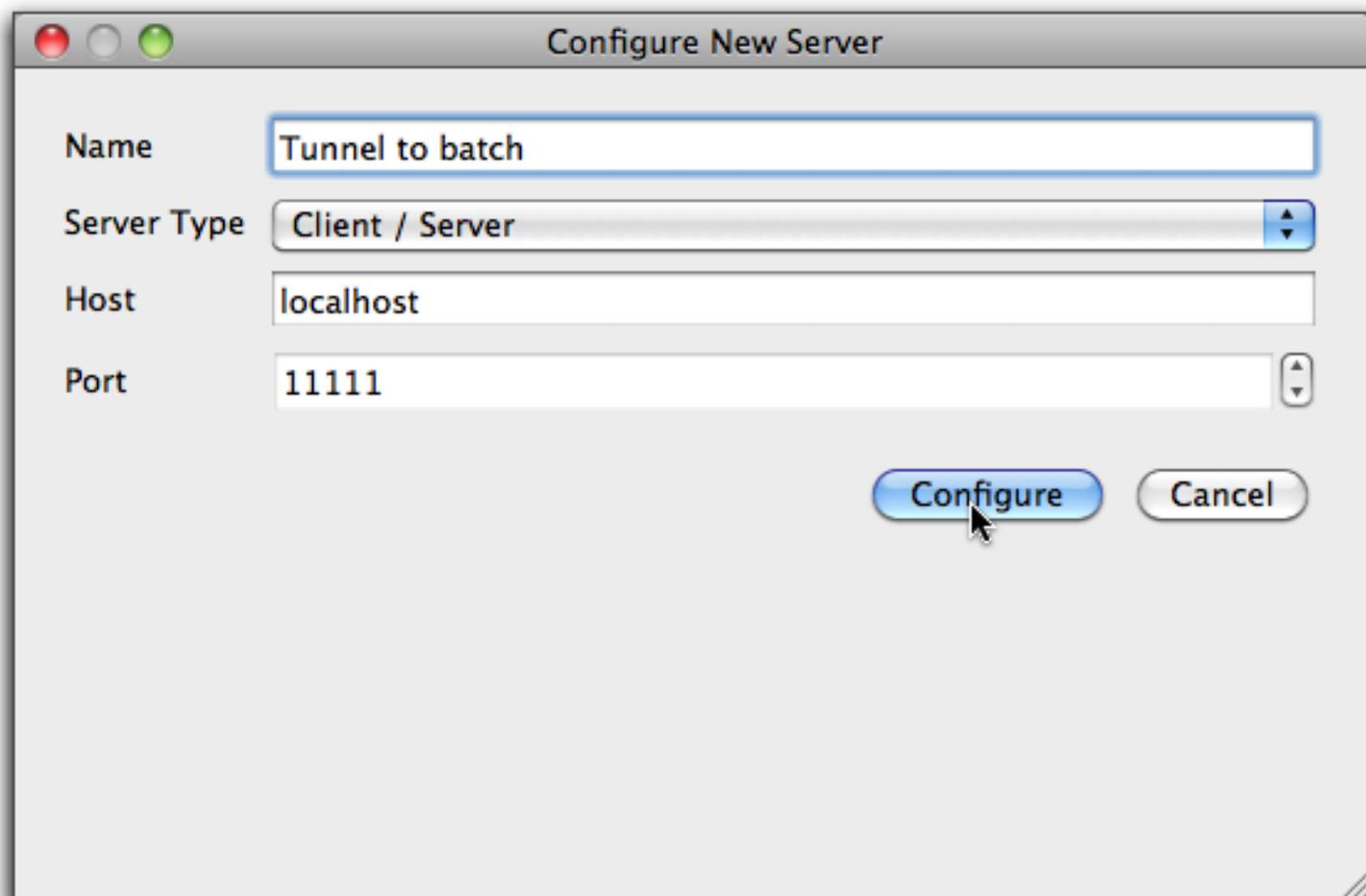


You are now presented with the **Choose Server** dialog.

If you already have a server called "Tunnel to batch," then select it and click "Connect." Skip to "**Test to verify parallel connection**" below or just get to work.

If not, then click on "Add Server" and proceed to the next step.

Configure the server

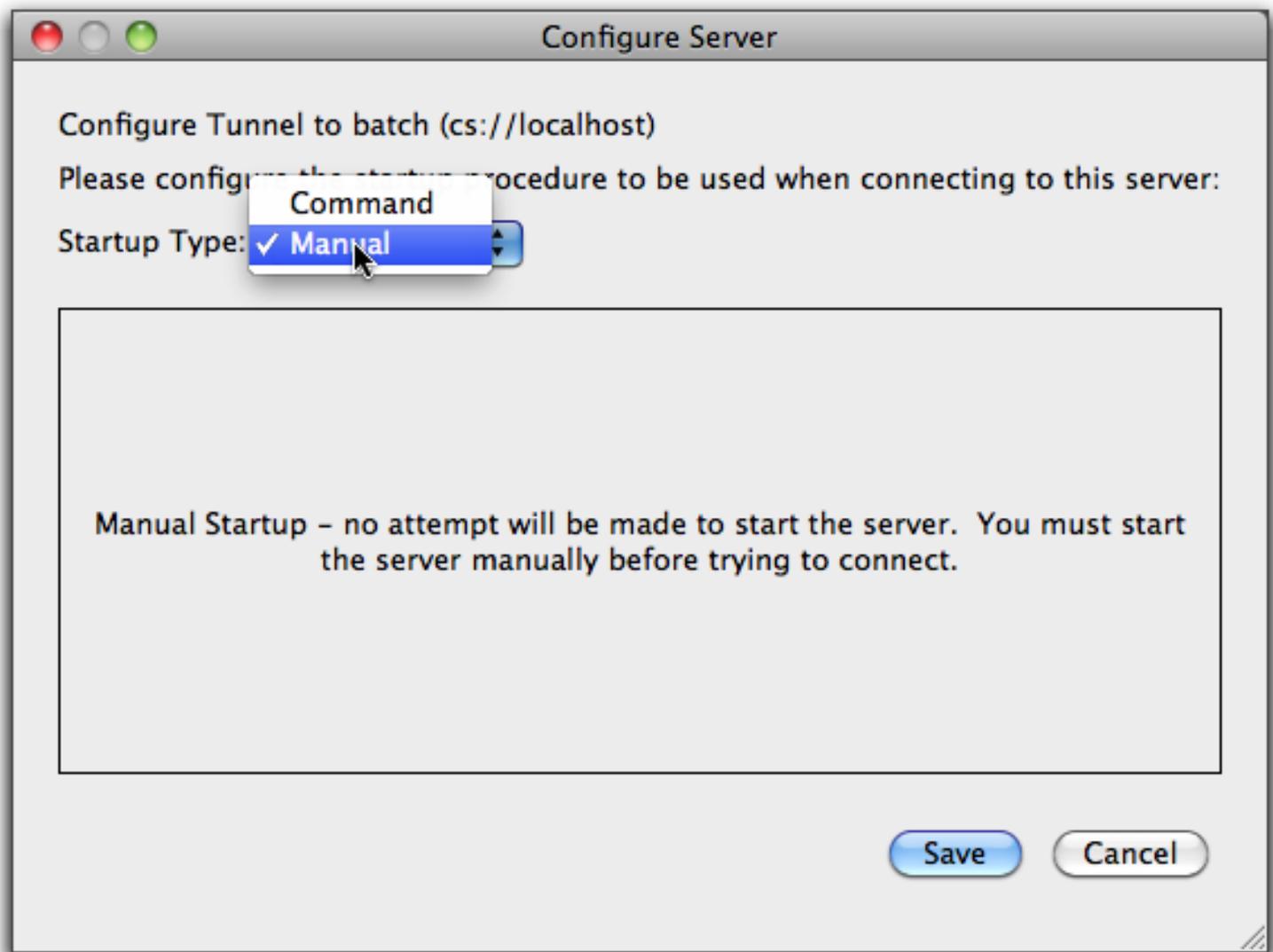


Give the new server a name, like "Tunnel to batch" or "Connect to batch." Then click the "Configure" button.

For Scenario A or B: You are connecting to "localhost" in order to use the ssh tunnel you created above, so put "localhost" in the "Host" field.

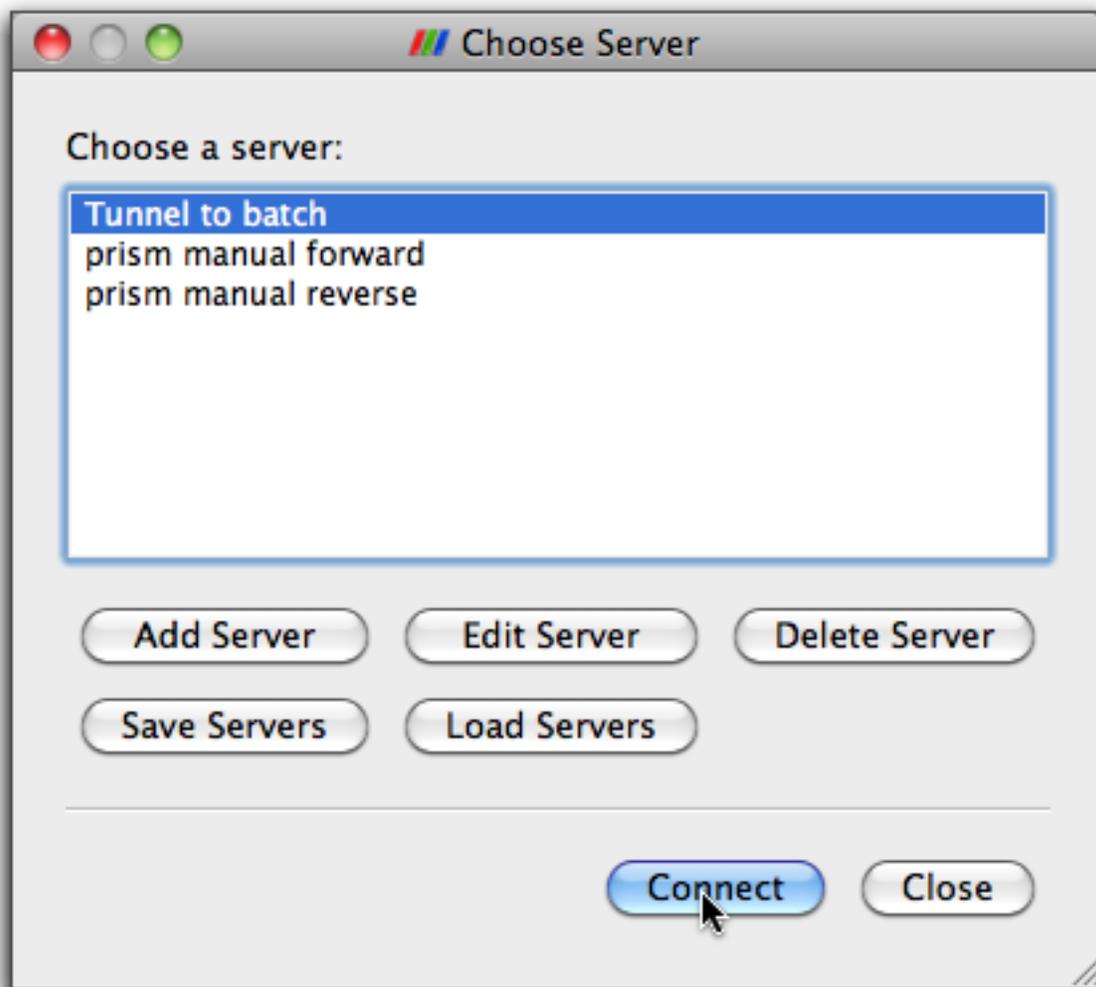
For Scenario C: Since you can see the batch nodes from your desktop, instead of "localhost", put the name or IP address of the first node in your batch job in the "host" field.

Setup server as manual



Choose "Manual" under "Configure Server" and then click the "Save" button.

Save and connect to the server



Recommended: Click "Save Servers" if you want your new server to be around next time you launch Paraview.

Choose the server you just created and click the "Connect" button. The batch xterm should give the message "client connected" and the "Choose Server" dialog should disappear. Congratulations. You are now running ParaView in parallel! You can browse for your data on the remote host and visualize it. Don't forget to hit the Apply button in ParaView after opening your data!

Test to verify parallel connection

The screenshot shows the ParaView 3.14.0 64-bit interface. The main window displays a 3D visualization of a protein structure (1cdw.pdb) with Process ID Scalars applied, showing the structure divided into colors (blue, red, orange) representing different processors. The Pipeline Browser on the left shows the dataset '1cdw.pdb' and the filter 'ProcessIdScalars1'. The Properties panel below the Pipeline Browser shows the 'ProcessIdScalars1' filter settings. The Statistics Inspector at the bottom shows the following data:

Name	Data Type	No. of Cells	No. of Points	Memory (MB)	Color
1cdw.pdb	Polygon...	8700	9404	0.456	Un...
ProcessIdSc...	Polygon...	2175	2357	0.133	0.1

The Memory Inspector panel on the right shows the following data:

Process	Memory Used	Memory Free
paraview	10.14 GBs	13.86 GBs
pvserver : 0	1.60 GBs	29.63 GBs
pvserver : 1	1.61 GBs	29.63 GBs
pvserver : 3	1.74 GBs	29.50 GBs
pvserver : 2	1.71 GBs	29.52 GBs

If you apply the "Process ID Scalars" filter (under the Filters menu) to your dataset you will then see that it is divided into colors, one for each processor being used to view the data. This is a good way to confirm it is all working correctly. Another way to see information about the servers is to choose View->Memory Inspector. Both methods are shown in the above image.