Paraview Client & Server

Timothy Brown
June 25th 2015

Paraview can be run in a client-server mode.

In this tutorial we will run the Paraview GUI (client) on your desktop and `pvserver` (server) on Yeti.

We will be using a ssh tunnel for the communication between the Paraview client and server.

Running the Paraview GUI (client) on your desktop means all the rendering is still done using your desktop graphics processor. This will be a bottleneck if you are trying to visualize huge datasets.

SSH

1. Log in to Yeti.

    ```
    laptop ~$ ssh yeti.cr.usgs.gov
    ```


2. Start a compute job.

    ```
    login01 ~$ salloc -A training -p prod \
                      -t 01:00:00 -N 1    \
                      -exclusive
    ```

3. Start the paraview server.

```
compute80 ~$ module load viz/paraview-4.3.1-50
compute80 ~$ module load mpi/mpich-x86_64
compute80 ~$ mpiexec -np 12 pvserver \
                     --use-offscreen-rendering
```

The pvserver will bind to port 11111 by default.

4. Create a tunnel.

```
laptop ~$ ssh -f -N -L 11111:compute80:11111 \
              yeti.cr.usgs.gov
```
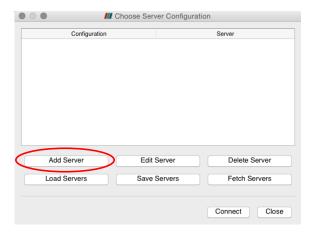

The tunnel takes the form of

    -L local_port:remote_host:remote_port

So we are forwarding the default Paraview port (11111) on our laptop to the compute node port 11111, through the login node.
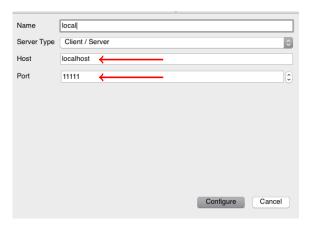
5. Start paraview locally.

   `laptop ~$ paraview`
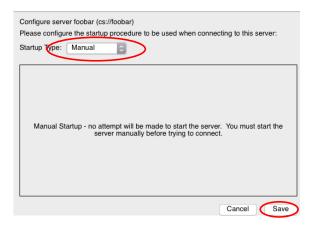
6. Create a new connection.

7. Add a new server.
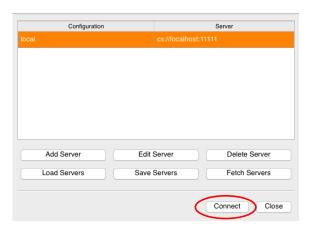
8. New server definitions.
   8.1 It can have any unique name.
   8.2 Use the localhost (loop-back)
   8.3 Use the local port in the ssh tunnel we just defined.

Name	local
Server Type	Client / Server
Host	localhost ←
Port	11111 ←

Configure   Cancel

9. Use the manual startup procedure, since `pvserver` is already running on the compute node.

10. Connect to the new server.

License