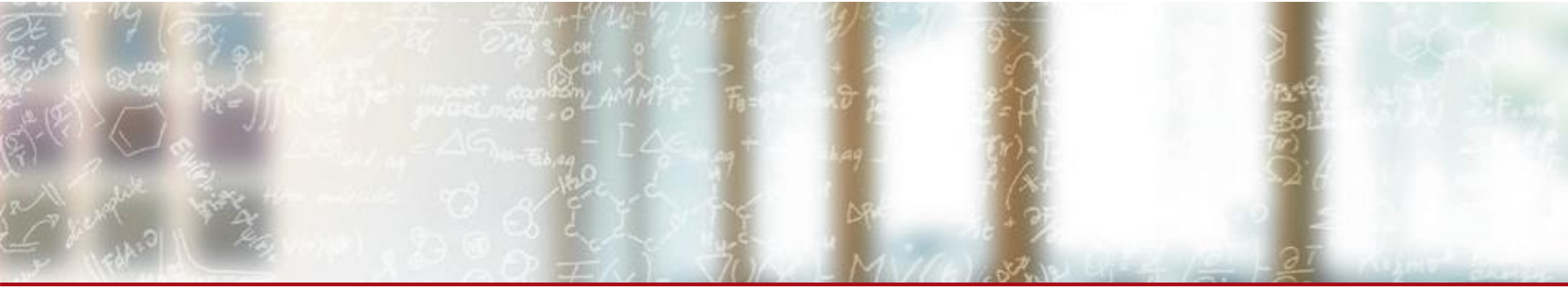
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



ParaView Client-Server on Piz Daint

Direct access to Piz Daint via ssh tunnel

Jean M. Favre, CSCS

November 6, 2020

Setup

- **Paraview provides an optimized communication between a remote (parallel) server and a desktop client. This is far more efficient than running ParaView inside a VNC desktop, or using ssh -X**
- **Access to your remote data is still enforced.**
- **A ParaView communication must be configured**
  - Client pre-compiled desktop version
  - Use ParaView server (pvserver) compiled on daint
  - Remote server configuration

Windows users

- **A Putty access must be configured**
  - PuTTY app
  - Private ssh key
    - Consult https://user.cscs.ch/access/auth/#generating-ssh-keys
    - Suggested reading
  - Putty Session for ParaView

Putty setup

- Create a new Putty Session (Next slide)

- Preliminary: You will need your userid number on daint:
- ssh to daint. The userid can be seen with the command "id".
- *For the rest of this document, we use userid=1100.*

- *Replace that number with your personal userid.*

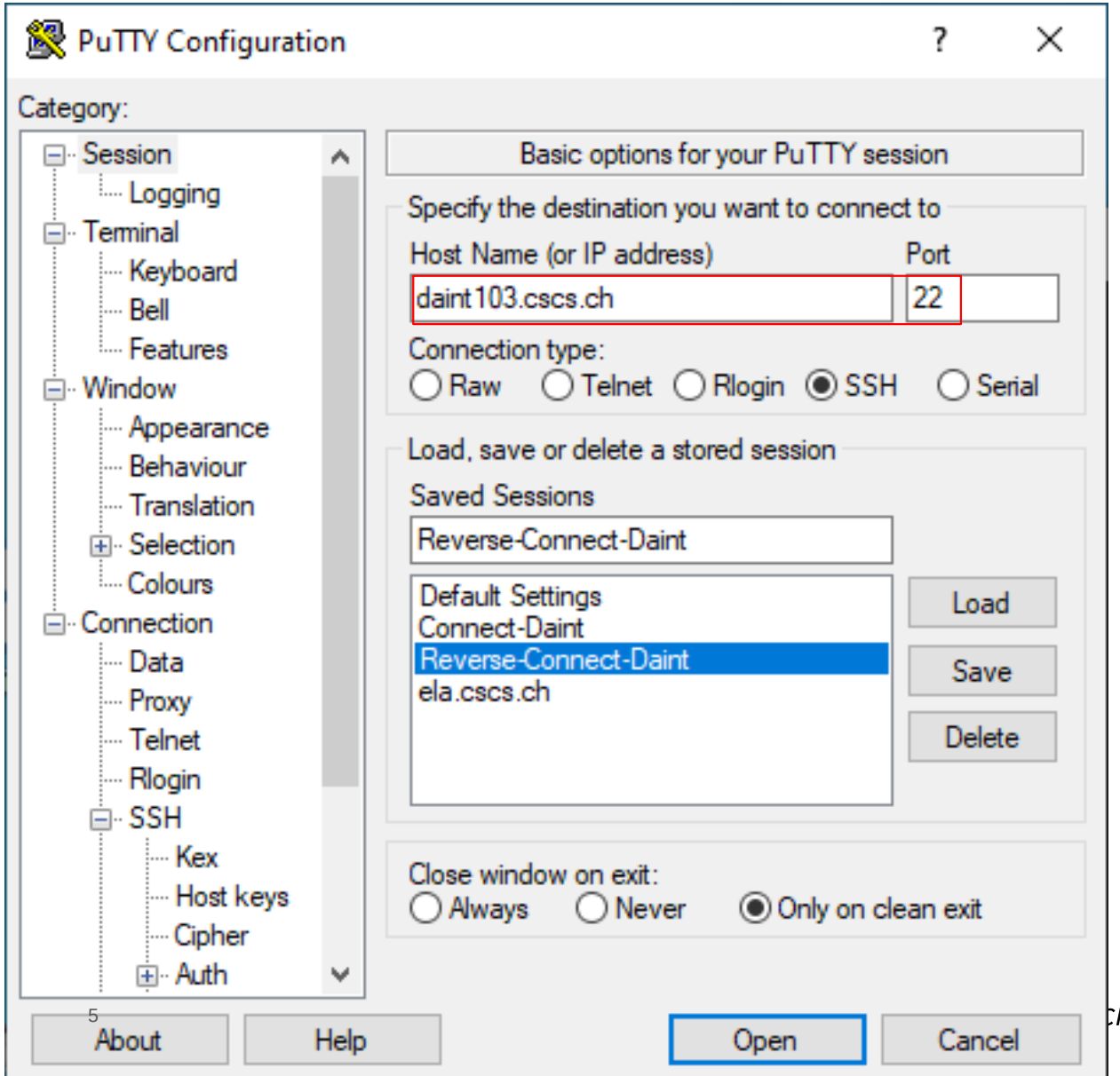
Under PuTTY Session:
New session with the destination system

In this example:

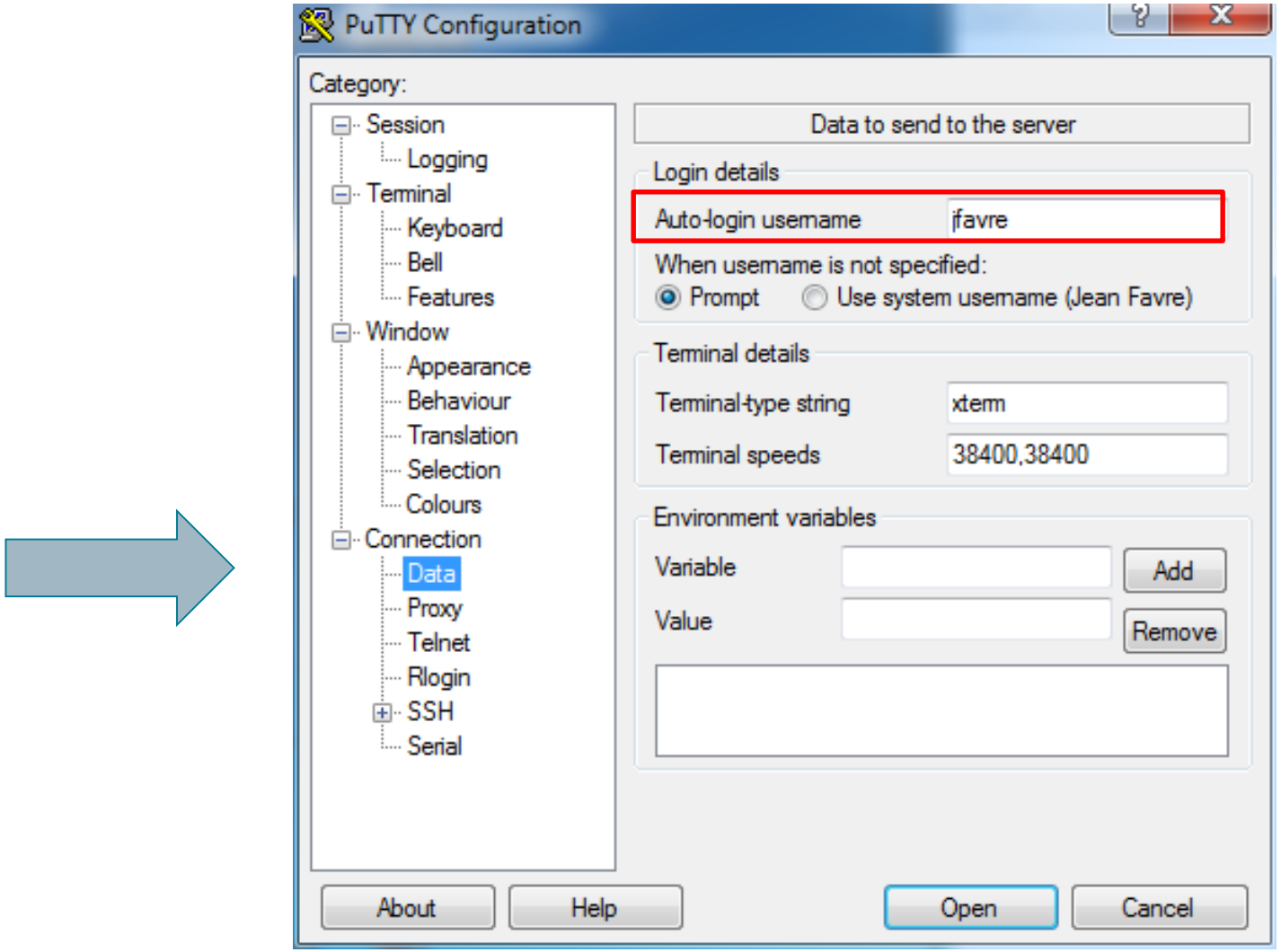
Access to daint.cscs.ch
is saved as PuTTY *session*

Reverse-Connect-Daint

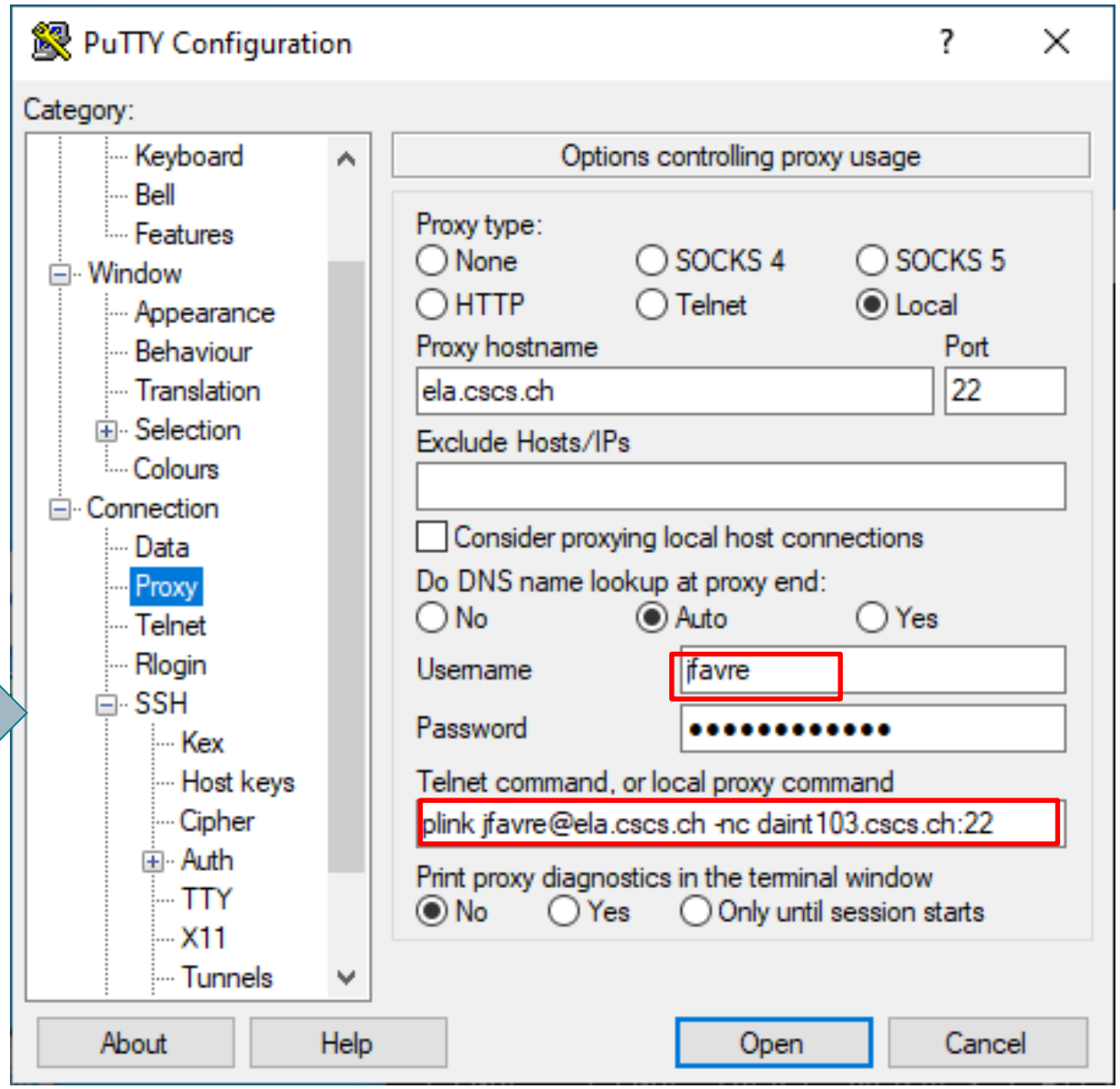
since ParaView uses a method called Reverse Connection, whereby the server (on daint) connects back to the client (on your desktop)



Under PuTTY Connection – Data:
Put your <username> on the auto-login tab.



Under PuTTY Connection – Proxy: Define the plink command.



Use the following command:

“plink <username>@ela.cscs.ch –nc daint103.cscs.ch:22”

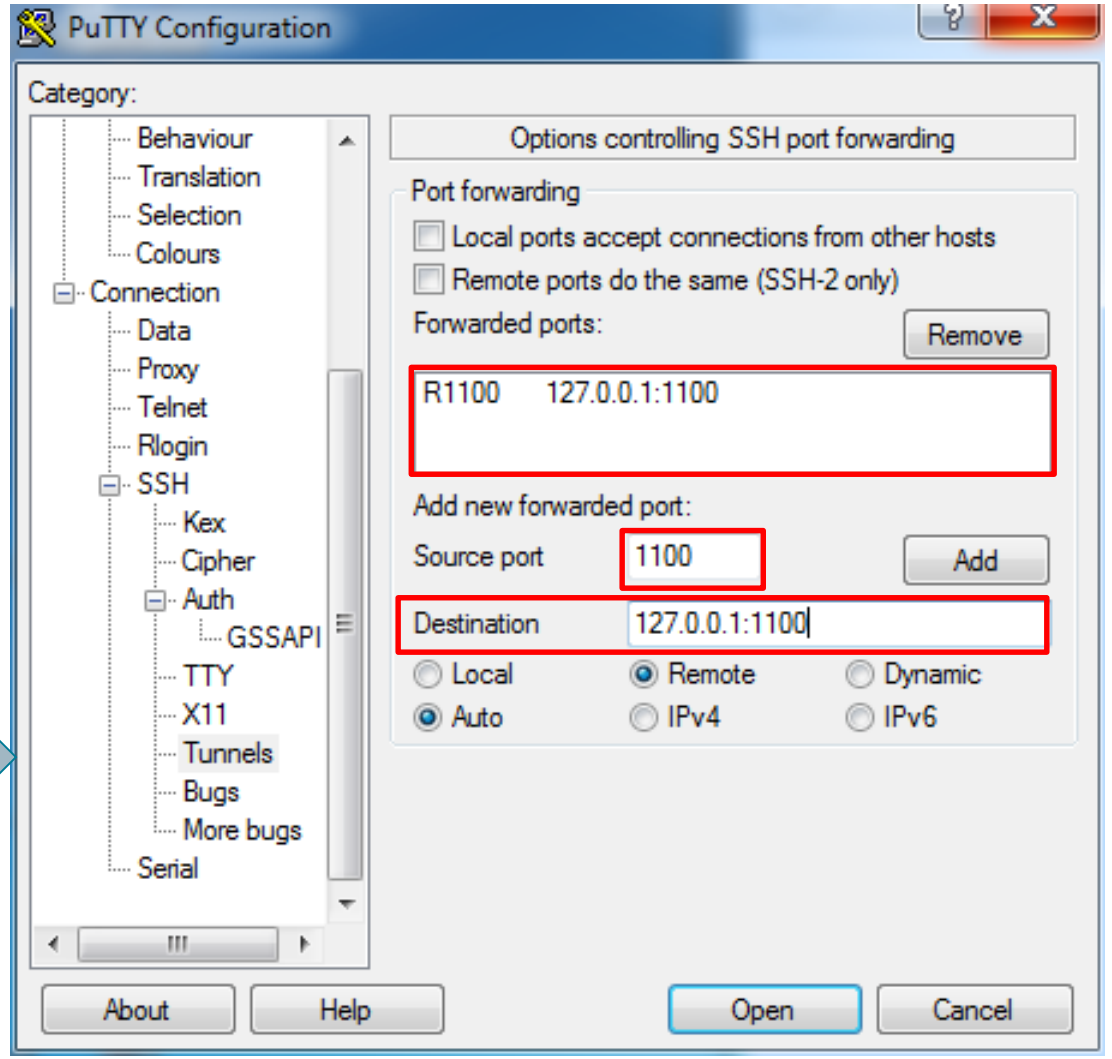
Under PuTTY Connection – SSH – Auth – Tunnels:
Define a single Remote tunnel from Daint

Select Remote, use your private userid number,

and define:

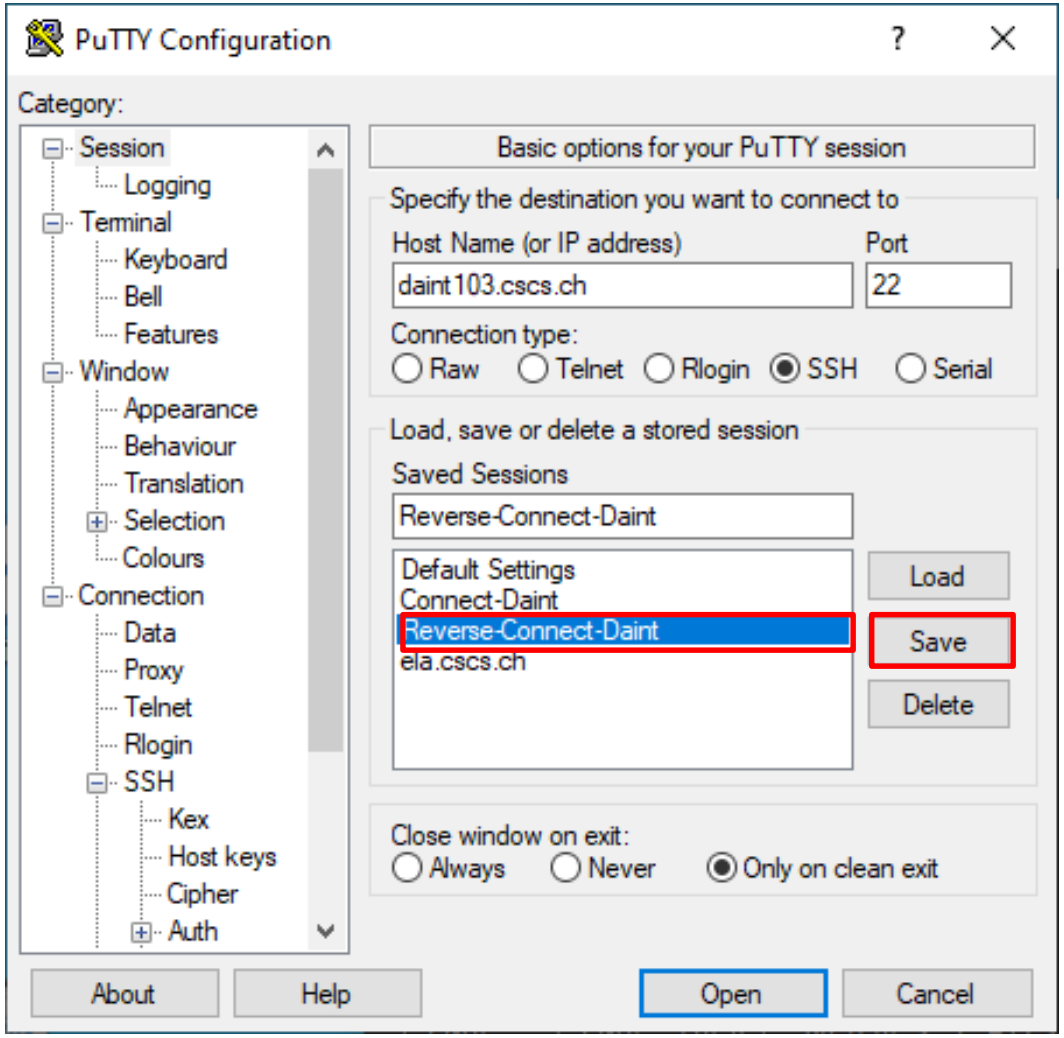
Source port = userid

Destination = 127.0.0.1:userid



Save PuTTY Session:

- *Call it „Reverse-Connect-Daint“*



ParaView setup I

- Two server configuration files for daint, available at:

- /apps/daint/UES/ParaView/server_daint_Windows.pvsc
- /apps/daint/UES/ParaView/server_daint.pvsc

- must be copied to your desktop and edited.

- Change the name “”jfavre” by your own username.

ParaView setup II

- Please note that the file “server_daint_Windows.pvsc” makes reference to another file "rc-submit-pvserver.sh".

- This is provided as a template. You should make a copy of this file to your remote private location, for example (on daint)

- cp /apps/daint/UES/ParaView/rc-submit-pvserver.sh $HOME

- This enables you to customize the shell script.

ParaView setup III

- edit “server_daint_Windows.pvsc” on your desktop and change the pathname of "rc-submit-pvserver.sh" inside it with the new pathname just created on daint.

- Follow up by editing the following lines:

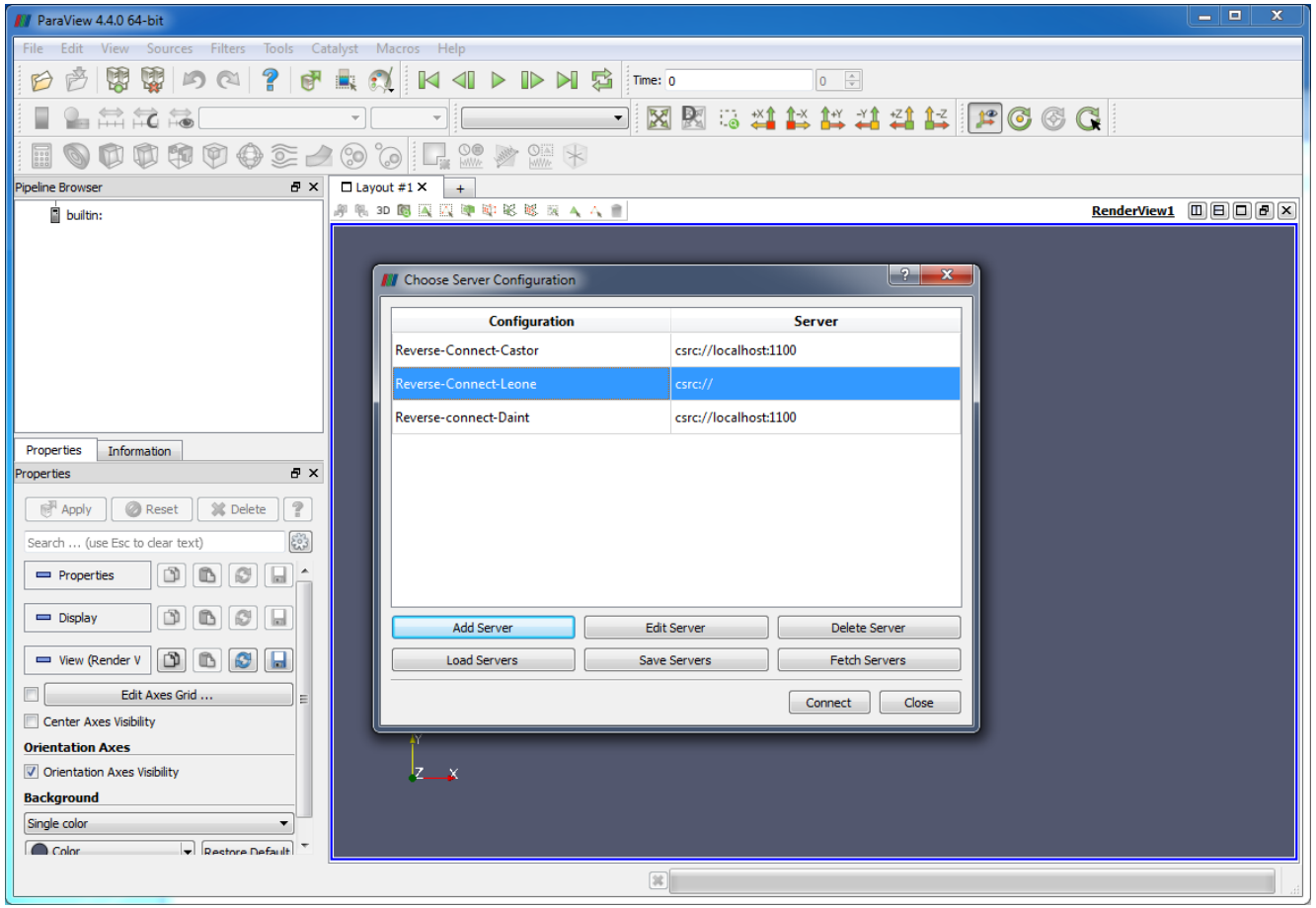
- <Option name="PV_SERVER_PORT" label="PV server port">
- <Range type="int" min="1024" max="65535" step="1" default="1234"/>

- the default value is set to 1234. This should be changed with your private Unix userid on daint.
- Save and Exit the editor.

ParaView setup IV

- Start ParaView
- Menu File->Connect->Load Servers and select file

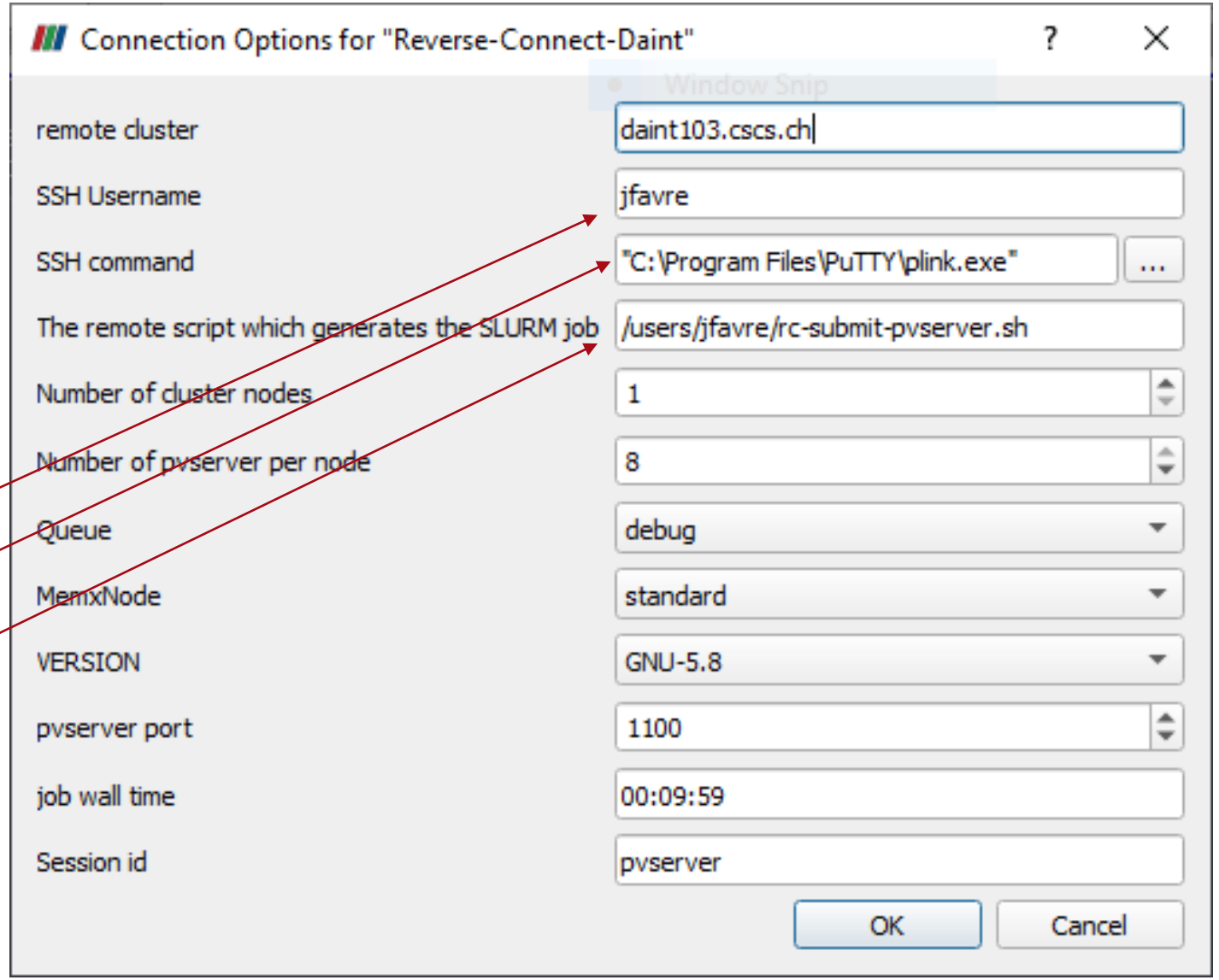
server_daint_Windows.pvsc



ParaView setup V

- Double-clicking on the configuration "Reverse-Connect-Daint", you will now be able to connect.
- Select options for your run.

- Be careful with:
  - Username
  - Double quotes ““
  - Path to script



Console output (example of what you should see)

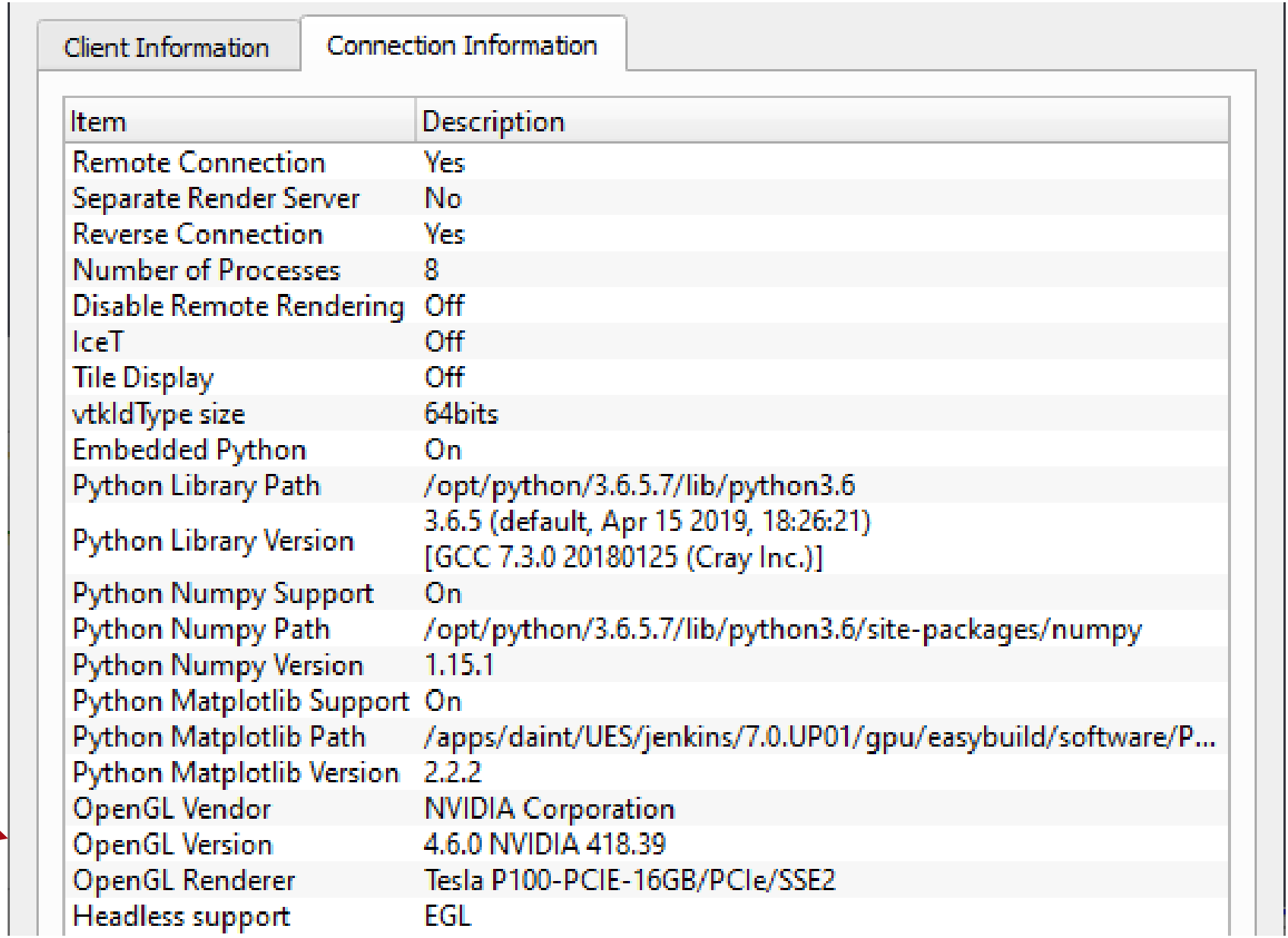
```
Accepting connection(s): rancate:11111
#SBATCH --job-name=pvserver
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH --ntasks=8
#SBATCH --time=00:19:59
#SBATCH --partition=normal
#SBATCH --constraint=gpu

srun -n 8 -N 1 --cpu_bind=sockets pvserver -rc -ch=daint103.cscs.ch -sp=11111
Submitted batch job  123456789
```

Sanity check

Are you connected to a remote parallel server?

Check menu Help-> About->connection information

gpu-partition with gpu-rendering

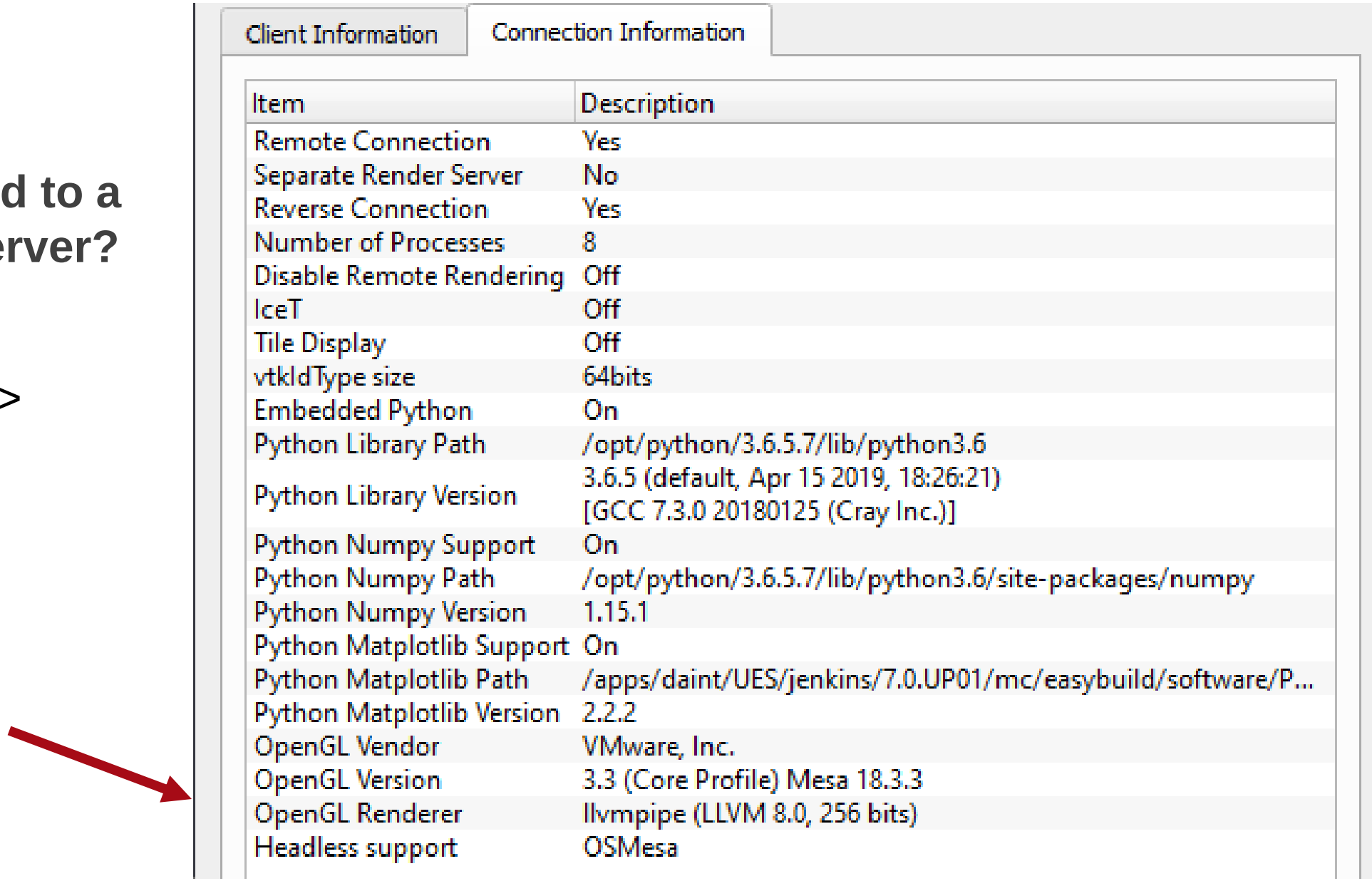


Sanity check

Are you connected to a remote parallel server?

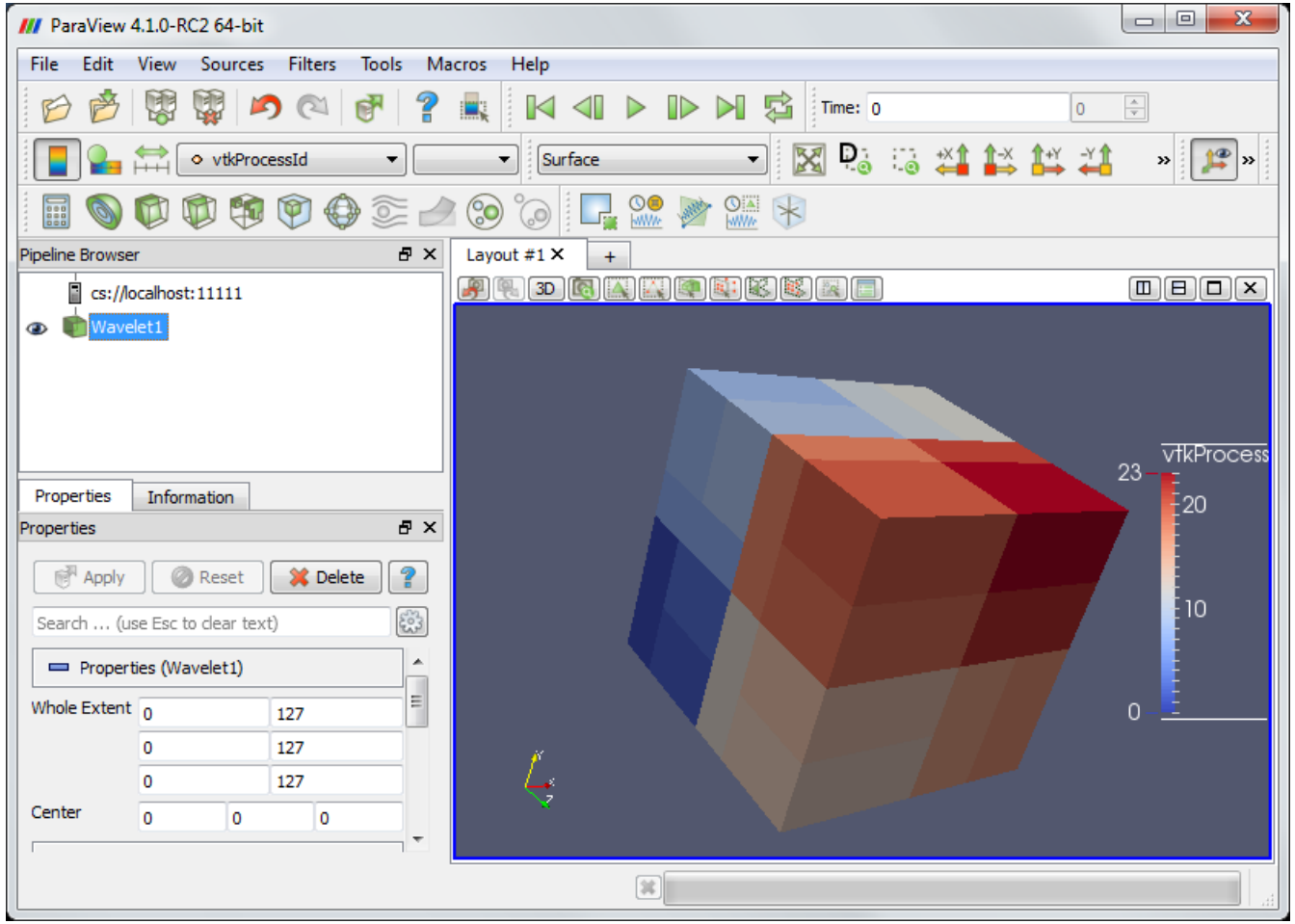
Check menu Help-> About->connection information

mc-partition with cpu-rendering



Are you [really] connected to a remote parallel server?

- Check connection and parallelism with a Wavelet source, displaying variable “vtkProcessId”



Manual connection without a GUI

Terminal 1

Use pvpython, or the python shell in ParaView

>> from paraview.simple import *

>> ReverseConnect("1100")

==========

Once connected:

- info = GetOpenGLInformation(location=servermanager.vtkSMSession.SERVERS)
- info.GetVersion() '4.6.0 NVIDIA 418.39'
- info = GetOpenGLInformation(location=servermanager.vtkSMSession.CLIENT)
- info.GetVersion() '4.5.0 NVIDIA 440.44'

Terminal 2

- LINUX users

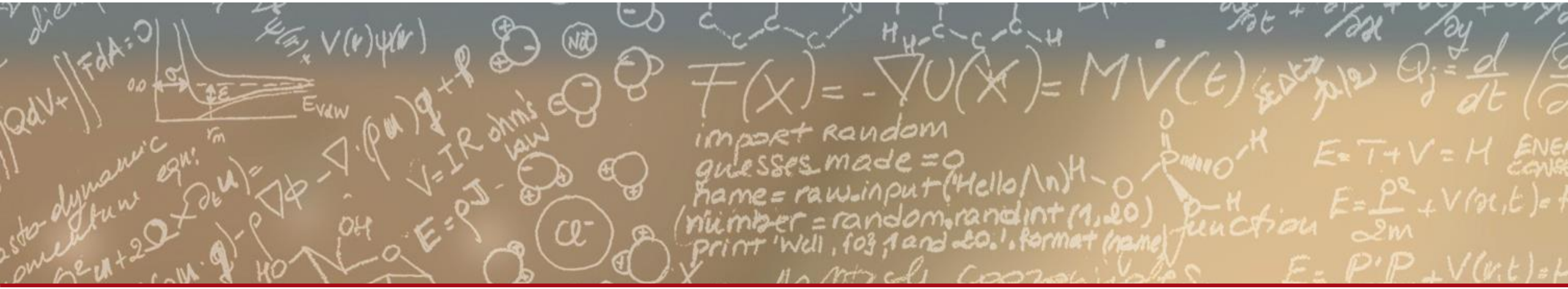
ssh -l jfavre -R 1100:localhost:1100 daint103.cscs.ch "/users/jfavre/rc-submit-pvserver.sh pvserver 00:29:59 1 2 1100 daint103.cscs.ch GNU-5.8 normal standard; sleep 6000”

==========

- Windows users

plink –load Reverse-Connect-Daint "/users/jfavre/rc-submit-pvserver.sh pvserver 00:29:59 1 2 1100 daint103.cscs.ch GNU-5.8 normal standard; sleep 6000”

==========


import random
guesses_made = 0
name = raw_input('Hello! What is your name?\n')
number = random.randint(1, 20)
print 'Well, ' + name + ', I am thinking of a number between 1 and 20.'
E = T + V = H
E = p²/2m + V(x,t)
E = P'P + V(x,t)
F(x) = -∇U(x) = MV(t)
V = IR
ohm's law