Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Remote Scientific Visualization at Jülich Supercomputing Centre

Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany
Cross-Sectional-Team Visualization

Visualization at JSC

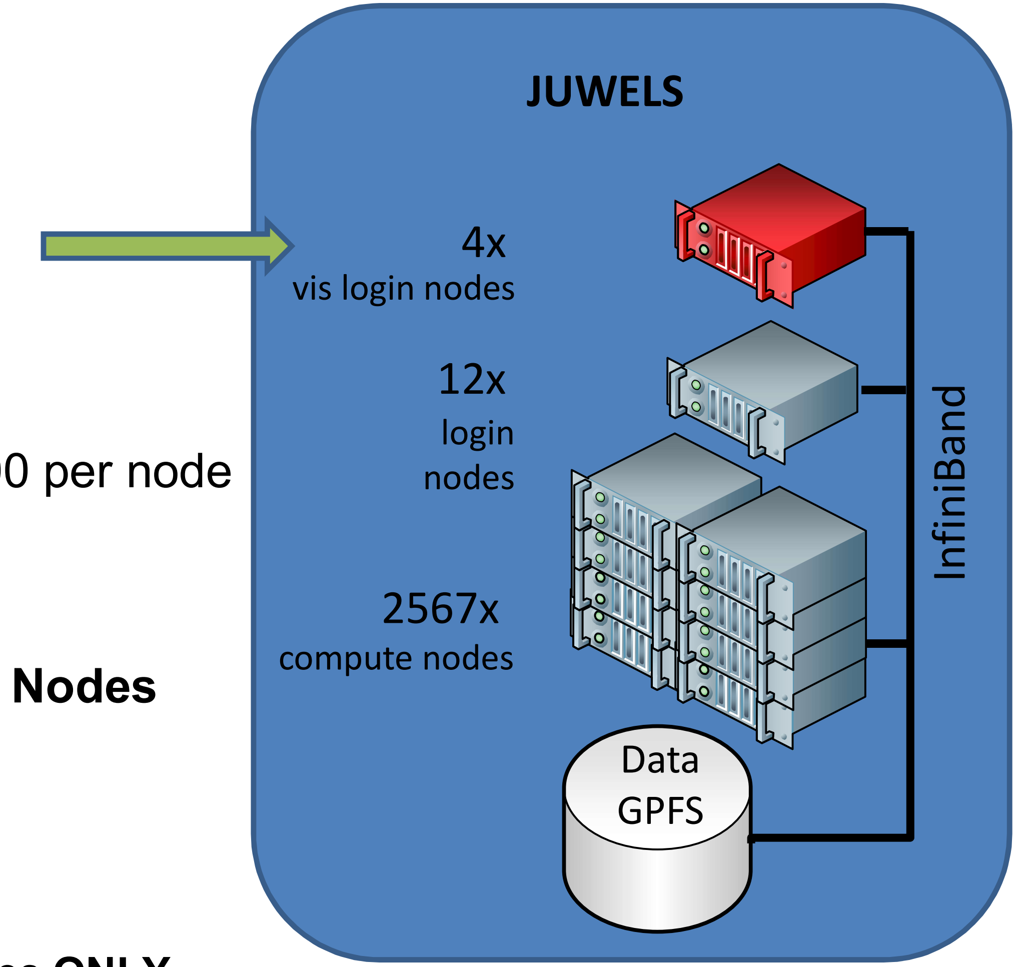
JUWELS: General Hardware Setup

4 x Visualization Login Nodes

- juwelsvis.fz-juelich.de
- (juwelsvis00 to juwelsvis03 in round-robin fashion)
- 768 GB RAM each

- 1 GPUs Nvidia Pascal P100 per node
- 12 GB RAM on GPU

No specific Visualization Batch Nodes



Keep in mind:

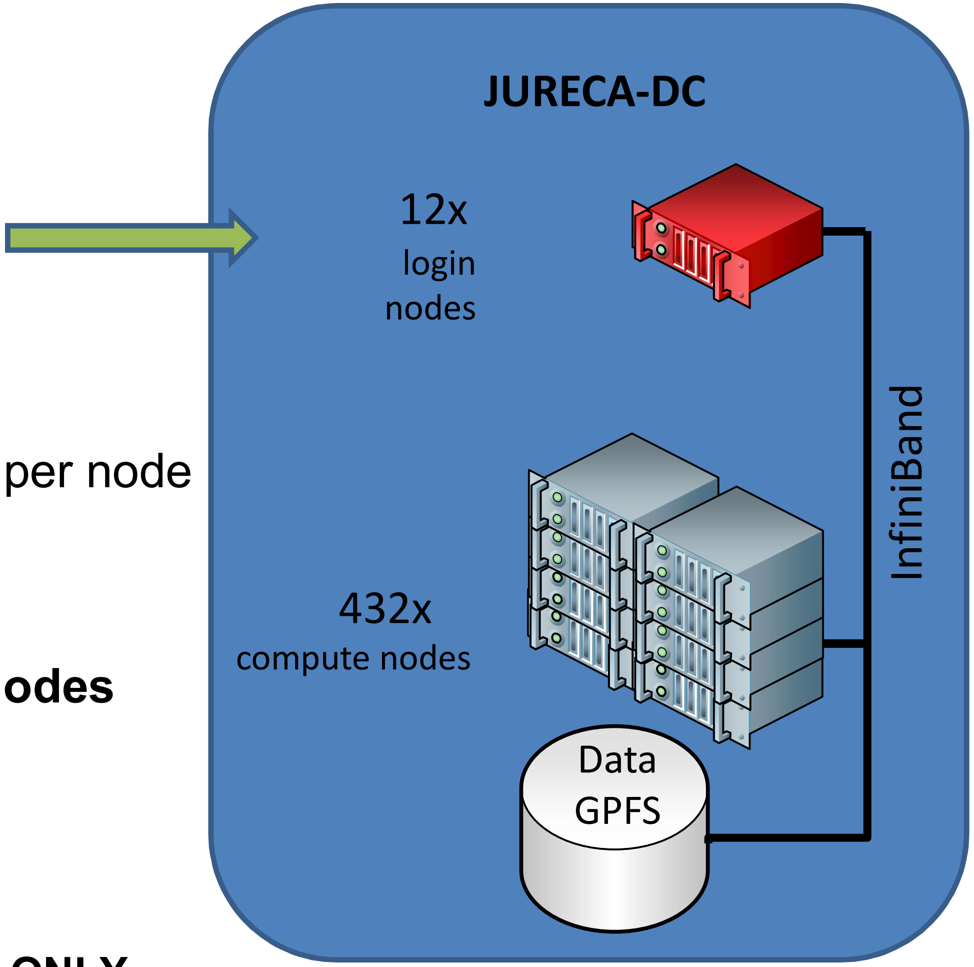
Visualization is **NOT** limited to vis. nodes **ONLY**.
(software rendering is possible on any node)

Visualization at JSC

JURECA-DC: General Hardware Setup

12 x Login Nodes with GPU

- jureca.fz-juelich.de
- (jureca01 to jureca12 in round-robin fashion)
- 1024 GB RAM each
- 2 x Nvidia Quadro RTX8000 per node
- 48 GB RAM on each GPU



No specific Visualization Batch Nodes

Keep in mind:

Visualization is **NOT** limited to vis. nodes **ONLY**.
(software rendering is possible on any node)

Visualization at JSC

General Software Setup

Special Software Stack on Vis Nodes:

Base Software:

- X-Server, X-Client (Window-Manager)
- OpenGL (libGL.so, libGLU.so, libglx.so), Nvidia

Middleware:

- Xpra
- Virtual Network Computing: VNC-Server, VNC-Client
- VirtualGL

Parallel and Remote Rendering Apps, In-Situ Visualization:

- ParaView
- VisIt

Other Visualization Packages (installation on user demand):

VMD, PyMol, Blender, GPicView, GIMP

Visualization at JSC

Usage Model for Vis Nodes

JUWELS projects:

- Visualization possible on 4 vis login nodes
- No specific visualization batch nodes
- JUWELS-Booster user have access to JUWELS vis login nodes

JURECA-DC projects:

- Visualization possible on all 12 Login nodes with 2x Nvidia RTX8000
- No specific visualization batch nodes
- As of December 2020, Visualization software stack under construction

Non HPC-Project Users:

- apply for test project

Mitglied der Helmholtz-Gemeinschaft

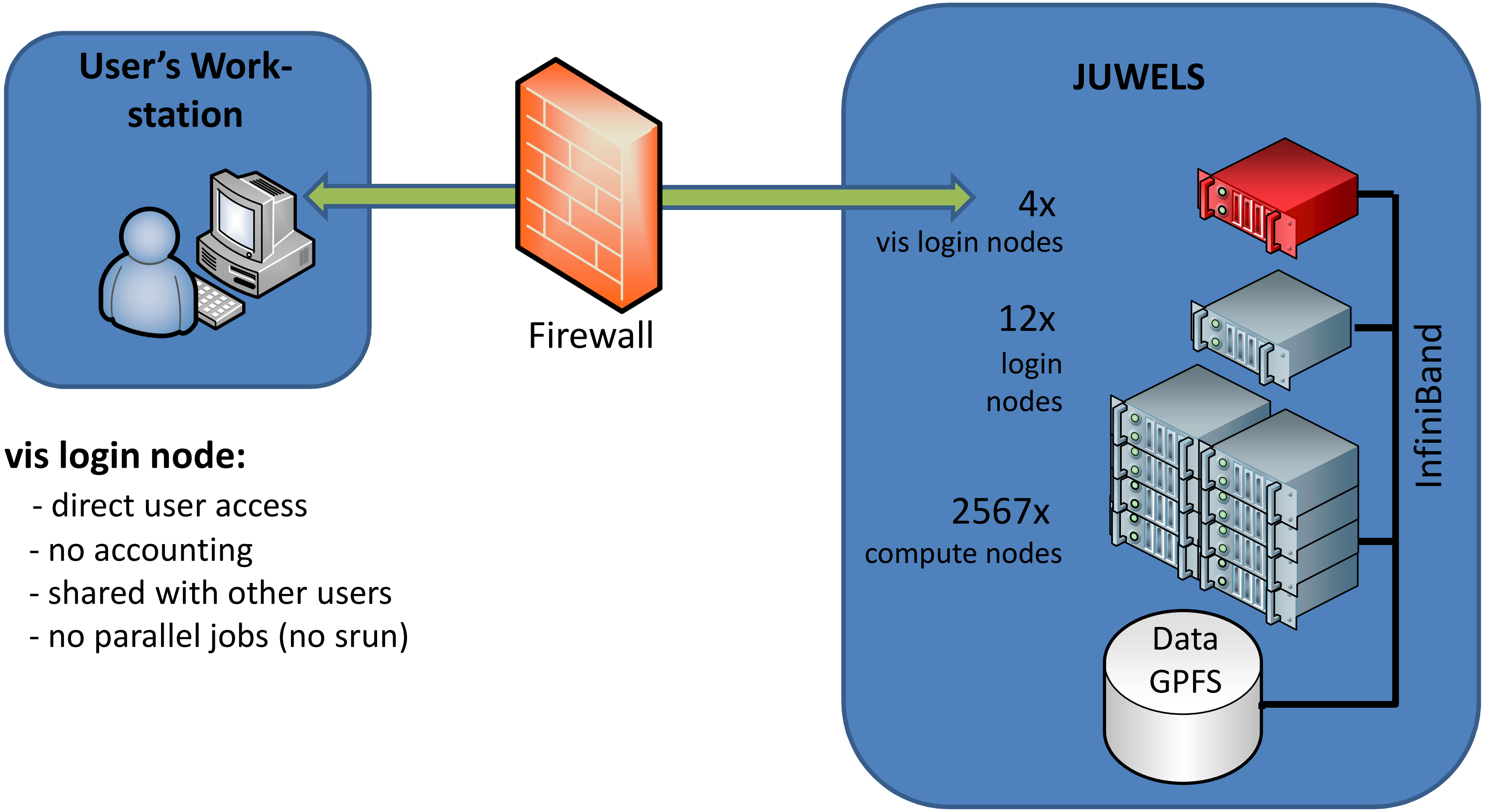
JÜLICH Forschungszentrum

Remote 3D Visualization

The following examples are given for JUWELS
Access to JURECA-DC similar

Remote 3D Visualization

General Setup



vis login node:

- direct user access
- no accounting
- shared with other users
- no parallel jobs (no srun)

Remote 3D Visualization

at Jülich Supercomputing Centre

- X forwarding + Indirect Rendering
  slow, maybe incompatible → bad idea

- “remote aware” visualization apps (ParaView, VisIt)
  application dependent error-prone setup

- Xpra - stream application content with H.264 + VirtualGL
  fast, our recommendation → good idea

- VNC (Virtual Network Computing) + VirtualGL
  full remote desktop, but slower than Xpra → medium good idea

Remote 3D Visualization

with X Forwarding + Indirect Rendering

Traditional Approach (X forwarding + Indirect Rendering)

ssh –X <USERID>@<SERVER>

- uses GLX extension to X Window System
- X display runs on user workstation
- OpenGL command are encapsulated inside X11 protocol stream
- OpenGL commands are executed on user workstation

- **disadvantages**
  - User´s workstation requires a running **X server**.
  - User´s workstation requires a **graphic card** capable of the required OpenGL.
  - User´s workstation defines the **quality and speed** of the visualization.
  - User´s workstation requires **all data needed** to visualize the 3d scene.
  - This approach is known to be error prone (OpenGL version mismatch, …)

Try to AVOID for 3D visualization.

Remote 3D Visualization

with Xpra (or VNC) + VirtualGL

- X-applications forwarded by Xpra (or VNC) appear on the local desktop as normal windows
- allows disconnection and reconnection without disrupting the forwarded application

- **advantages**
  - **No X is required** on user´s workstation (X display on server).
  - **No OpenGL is required** on user´s workstation (only images are send).
  - Quality of visualization does **not depend** on user´s workstation.
  - Data size send is **independent** from data of 3d scene.
  - Disconnection and reconnection possible.
- **VirtualGL** for hardware accelerated rendering: use `vglrun <application>`
  - it **intercepts the GLX** function calls from the application and **rewrites them**.
  - The corresponding GLX commands are then sent to the X display of **the 3d X server**, which has a 3D hardware accelerator attached.

- Good solution for any OpenGL application e.g. ParaView, VisIt, IDL, VMD, PyMol, …

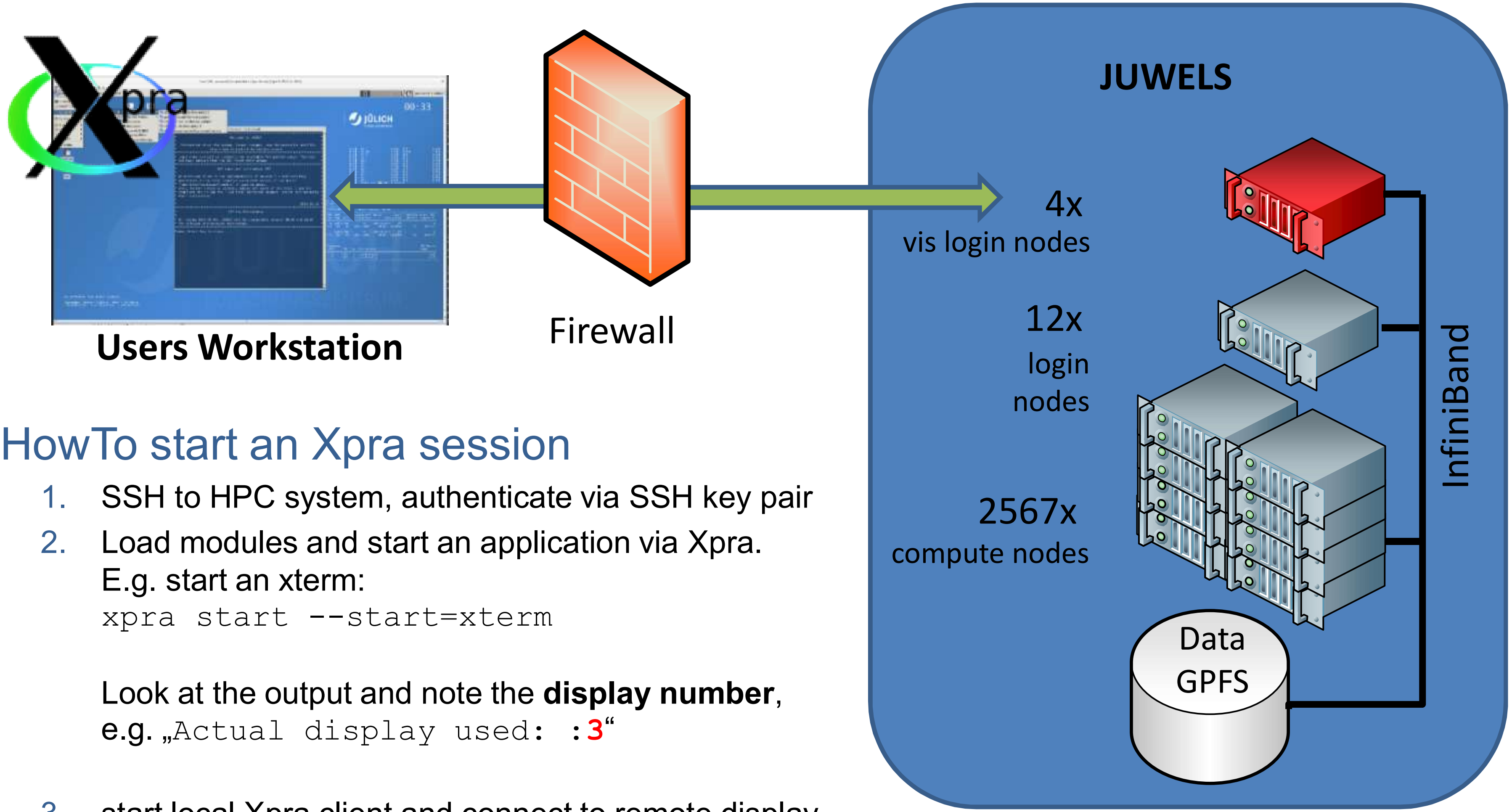
https://xpra.org/

Xpra Integration in JupyterLab@JSC

- How to start Xpra-Session:
  - **Within JupyterLab@JSC** https://jupyter-jsc.fz-juelich.de
    Brand New Feature: start Xpra and visualization apps from Jupyter in the Browser → to be presented in slides about JupyterLab (Jens Henrik Göbbert)
  - Alternative: start session manually, see next slides

Remote 3D Visualization

with Xpra + VirtualGL



HowTo start an Xpra session

1. SSH to HPC system, authenticate via SSH key pair
2. Load modules and start an application via Xpra.
   E.g. start an xterm:
   ```
   xpra start --start=xterm
   ```
   Look at the output and note the **display number**,
   e.g. „`Actual display used: :3`“
3. start local Xpra client and connect to remote display
4. Start visualization application in the xterm
5. Stop the Xpra session by `xpra stop :3`

Setup Xpra

Step 1: login to a (visualization) login node

- **Linux:**
  ```
  ssh <USERID>@juwelsvis02.fz-juelich.de
  ```
- **Windows:**
  connect via a ssh client, e.g. PuTTY. The PuTTY ssh keyagent pageant may be usefull, too.

Setup Xpra

Step 2: start xpra on HPC node and notice the display-number in the output

For example, start an xterm:

```
jwvis02> module --force purge
jwvis02> module use otherstages
jwvis02> ml Stages/Devel-2020  GCCcore/.9.3.0 xpra/4.0.4-Python-3.8.5

jwvis02> xpra start --start=xterm
...
Actual display used: :3
```

- The display-number is needed to connect to the Xpra session

Setup Xpra

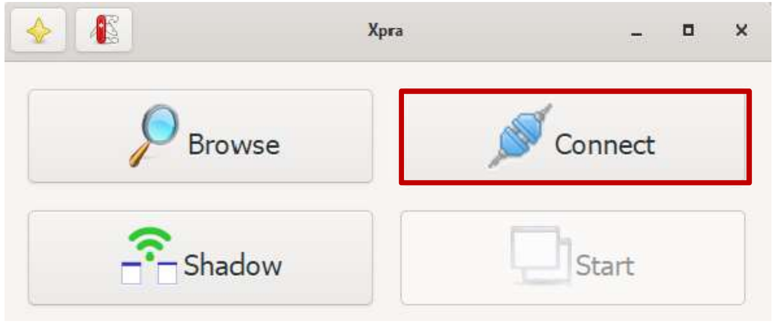
Step 3: connect to Xpra session

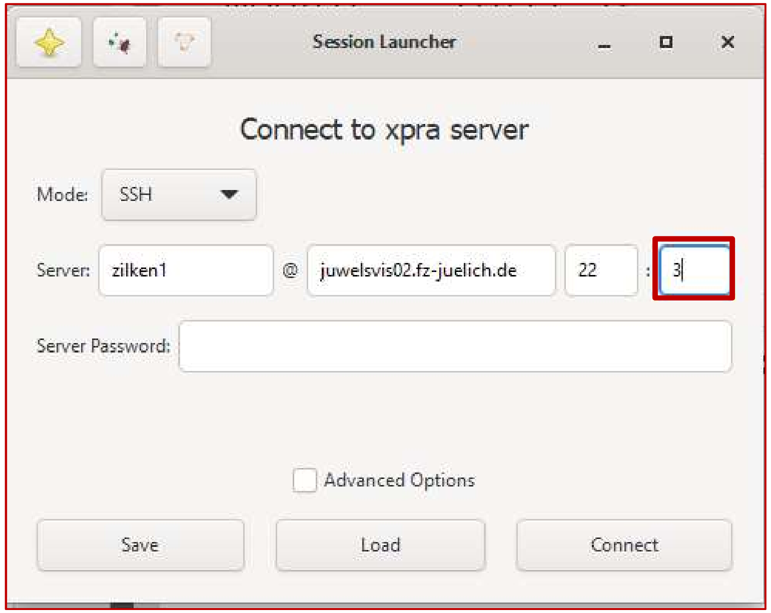
Install Xpra on your local machine. Download from www.xpra.org

Linux: use command

```
local_machine> xpra attach
ssh://USERNAME@juwelsvis02.fz-juelich.de/3
```

Windows: use Xpra GUI:





Connect to xpra server

Mode: SSH

Server: zilken1 @ juwelsvis02.fz-juelich.de 22 : 3

Server Password:

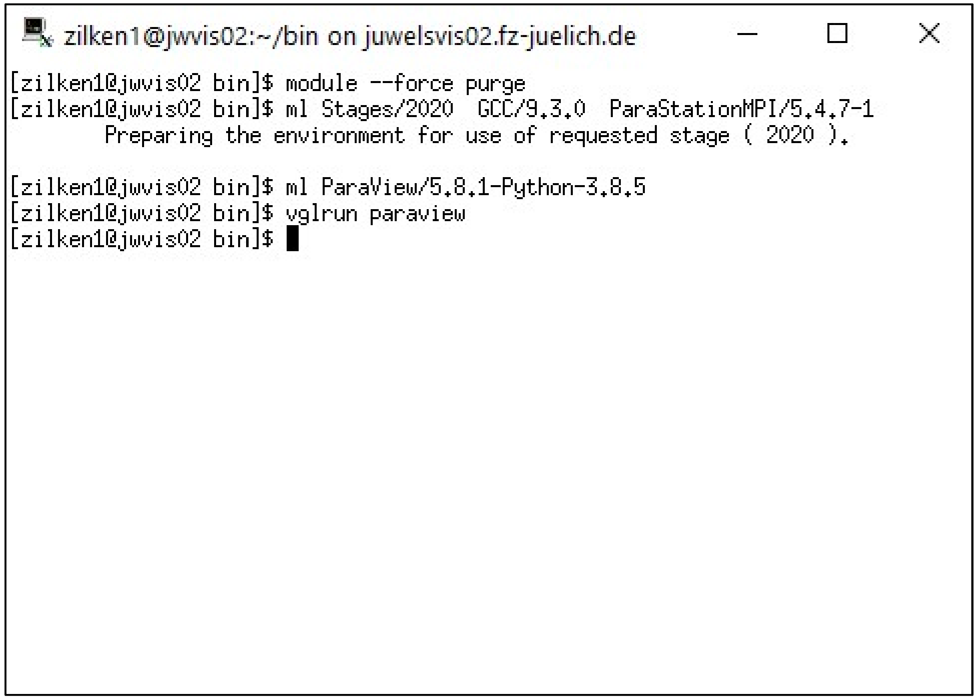
□ Advanced Options

Setup Xpra

Step 4: start visualization application

After successful connection, an xterm window will show up on your local desktop.

Start your application there, e.g. ParaView:



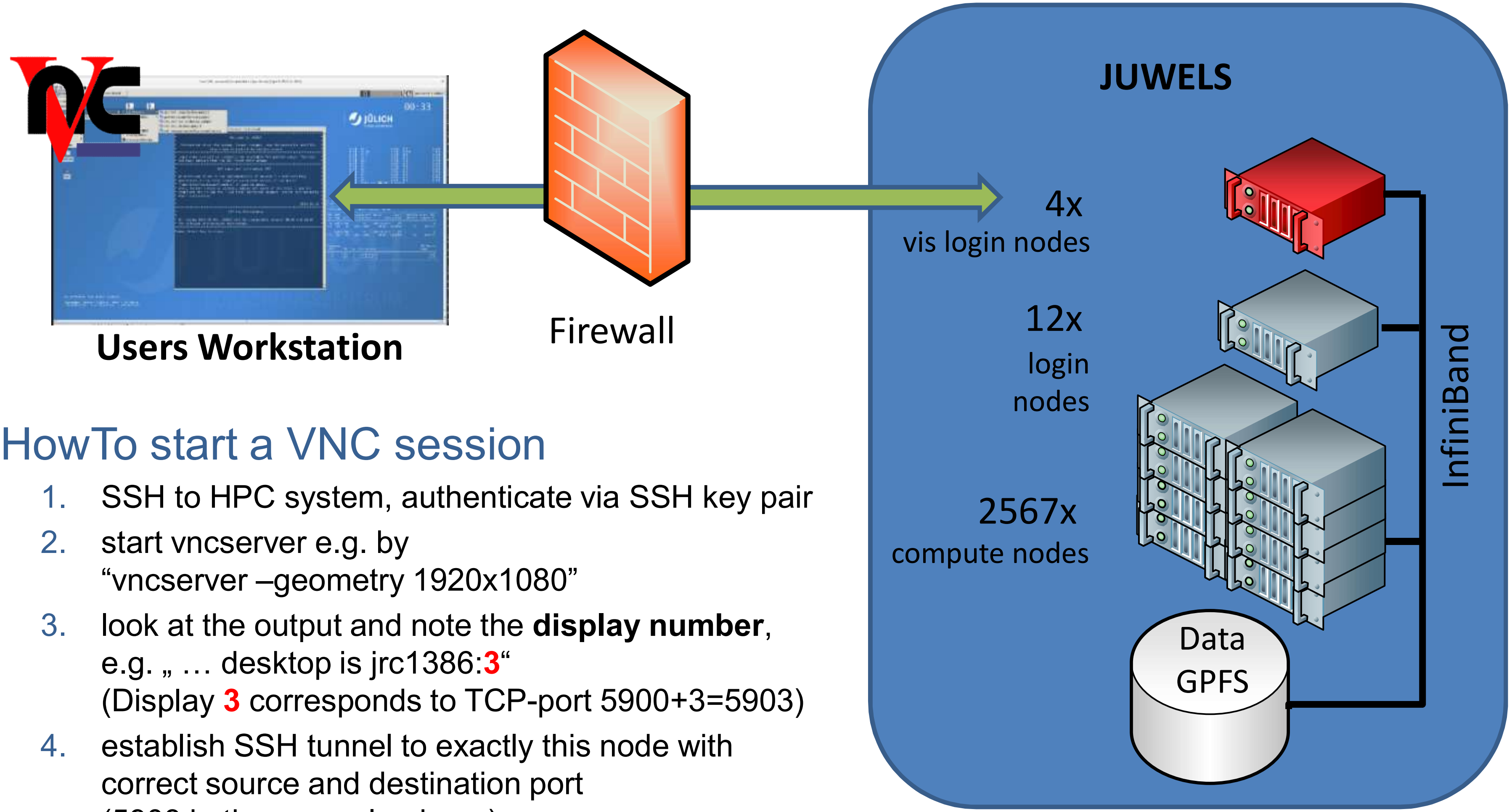
```
[zilken1@jwvis02 bin]$ module --force purge
[zilken1@jwvis02 bin]$ ml Stages/2020  GCC/9.3.0  ParaStationMPI/5.4.7-1
        Preparing the environment for use of requested stage ( 2020 ).

[zilken1@jwvis02 bin]$ ml ParaView/5.8.1-Python-3.8.5
[zilken1@jwvis02 bin]$ vglrun paraview
[zilken1@jwvis02 bin]$ 
```

Step 5: When you are done, stop the session by

```
jwvis02> xpra stop :3
```

Remote 3D Visualization

with VNC + VirtualGL



Users Workstation

Firewall

JUWELS

4x
vis login nodes

12x
login
nodes

2567x
compute nodes

Data
GPFS

InfiniBand

HowTo start a VNC session

1. SSH to HPC system, authenticate via SSH key pair
2. start vncserver e.g. by
   “vncserver –geometry 1920x1080”
3. look at the output and note the **display number**,
   e.g. „ … desktop is jrc1386:**3**“
   (Display **3** corresponds to TCP-port 5900+3=5903)
4. establish SSH tunnel to exactly this node with
   correct source and destination port
   (5903 in the example above)
5. start local VNC client and connect to remote display

https://trac.version.fz-juelich.de/vis/wiki/vnc3d/manual

Setup VNC Connection

Preliminary step: **setup a VNC Password**
(need only be done once)

- Login to a JUWELS vis login node or JURECA login node, create the directory `~/.vnc` and define VNC password
- E.g.:

```
ssh <USERID>@jurecavis.fz-juelich.de
mkdir ~/.vnc
vncpasswd
```

Setup VNC Connection

Example for JUWELS. Similar for JURECA, just use login nodes

Step 1: login to a specific visualization login node

- Hint: to establish a ssh tunnel, you need to connect to the same login node twice! Therefore:
  Don‘t use the „generic“ names (juwelsvis, jureca).
  Instead select a specific node randomly
  (juwelsvis00 .. juwelsvis03, jureca01 .. jureca12)

- **Linux:**
  ```
  ssh <USERID>@juwelsvis02.fz-juelich.de
  ```
- **Windows:**
  connect via a ssh client, e.g. PuTTY. The PuTTY ssh keyagent pageant may be usefull, too.

Setup VNC Connection

Step 2: start VNC-server on HPC node and locate the display-number in the output

Example:

```
vncserver -geometry 1920x1080
...
desktop is <node-name>:3
...
```

- The display-number is needed to establish the ssh tunnel (see step 3).
  The VNC-server listens to TCP-port 5900+display-number (5903 in the example)

Setup VNC Connection

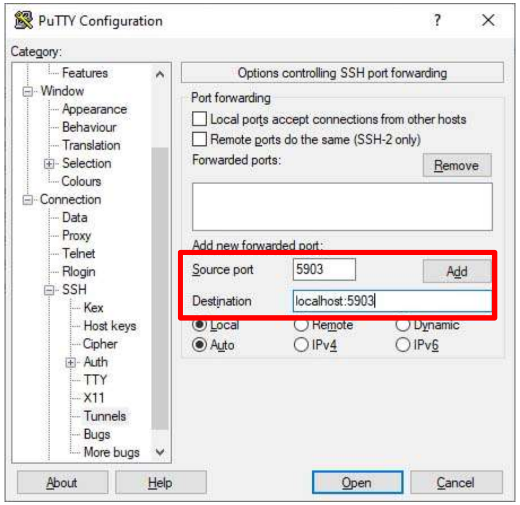
Step 3: establish the ssh tunnel

- Use the correct TCP port! Port must correspond to the display number (3 in this example)

- **Linux:**

```
ssh -N -L 5903:localhost:5903
<USERID>@juwelsvis00.fz-juelich.de
```

- **Windows:**
  Use e.g. PuTTY
  to setup the tunnel



PuTTY Configuration

Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

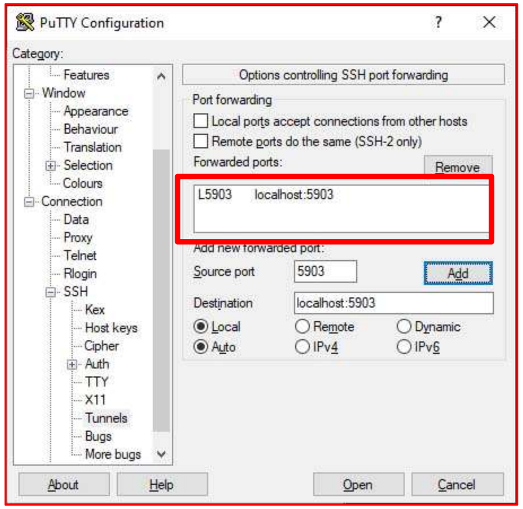
Add new forwarded port:

Source port: 5903

Destination: localhost:5903

● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6



PuTTY Configuration

Options controlling SSH port forwarding

Port forwarding

□ Local ports accept connections from other hosts

□ Remote ports do the same (SSH-2 only)

Forwarded ports:

L5903 localhost:5903

Add new forwarded port:

Source port: 5903

Destination: localhost:5903

● Local ○ Remote ○ Dynamic

● Auto ○ IPv4 ○ IPv6

Setup VNC Connection

Step 4: start your local VNC viewer

Linux:

VNC viewer typically is already part of the Linux distribution or can be installed from a repository. Just start vncviewer with the correct display-number:

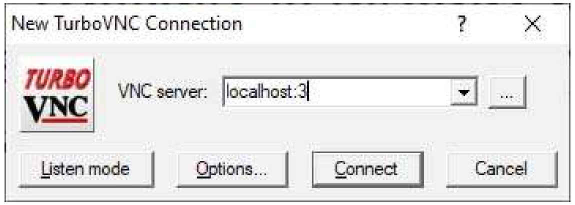
```
vncviewer localhost:3
```

Linux/Windows/Mac:

Download and install turboVNC:

https://sourceforge.net/projects/turbovnc/

Connect to localhost:3



New TurboVNC Connection

VNC server: localhost:3

Documentation

Visualization Related Documentation

Please visit

https://trac.version.fz-juelich.de/vis/

Please send us your feedback.

h.zilken@fz-juelich.de **j.goebbert@fz-juelich.de**