

Remote Visualization at JSC (with ParaView)

Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany Algorithms, Tools and Methods Lab Visualization & interactive HPC

Herwig Zilken, 2024/11/14



Algorithms, Tools and Methods Lab Visualization & interactive HPC

Scientific Visualization

 R&D + support for visualization of scientific data

Virtual/Augmented Reality

 VR visualization based on Unreal Engine, with head mounted displays and tablet computers for data analysis and presentation

Multimedia

 Multimedia productions for websites (e.g. Youtube) or presentations

Interactive HPC

- Jupyter@JSC
- Workflows



JUWELS: closer look at login nodes

Cluster

4 x Login Nodes with GPU

- juwelsvis00 to juwelsvis03
- (juwelsvis.fz-juelich.de)
- 768 GB RAM each
- 1 GPUs Nvidia Pascal P100 per node
- 12 GB RAM on GPU
- 9 x Login Nodes without GPU
 - juwels-cluster

Booster:

4 x Login Nodes without GPU

- juwels-booster
- no Xserver, no GPU →limited usage for visualization

Keep in mind: software rendering is possible on any node





JURECA-DC: closer look at login nodes

12 x Login Nodes with GPU

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

Keep in mind: software rendering is possible on any node





General Software Setup

Typical Software Stack for Visualization

Base Software:



X-Server, X-Client (Window-Manager)

OpenGL.

Middleware:



Xpra

Virtual Network Computing: VNC-Server, VNC-Client

VirtualGL (for remote hardware rendering, if possible)

OpenGL (libGL.so, libGLU.so, libglx.so), Nvidia or Mesa driver

Parallel and Remote Rendering App, In-Situ Visualization:

ParaView

ParaView



Vislt

Other Visualization Packages (more packages on user demand):

Blender, GPicView, VTK, VMD

JÜLICH Forschungszentrum

Remote 3D Visualization

at Jülich Supercomputing Centre

Bad

 X forwarding ("ssh −X") + indirect Rendering slow, maybe incompatible → bad idea

Medium good

 "intrinsic remote capable" visualization apps application dependent, error-prone setup

Our recommendation

- Remote Windows or Desktop
 - Xpra stream application content with H.264 + VirtualGL
 - Can be started with a single click in Jupyter@JSC
 For JURECA, Jupyterlab approach is buggy at the moment, don't use until bug is fixed → For now start manually on JURECA
 - VNC + VirtualGL can be used also
 - Simple, but full featured remote desktop
 - Does not be started from Jupyter@JSC

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

https://xpra.org/

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

How to use Xpra @ JSC



Two ways to start an Xpra session:

From JupyterLab@JSC

https://jupyter-jsc.fz-juelich.de

- Very easy setup, no additional software needed (only browser, but use Chrome, not Firefox!)
- Can be a little bit slow sometimes

Start Xpra session manually

- Can be a little bit faster than Xpra in browser
- Xpra client needs to be installed on your local machine
- Need to start Xpra on HPC system and locally by hand

Xpra Integration in JupyterLab@JSC



1. Go to https://jupyter-jsc.fz-juelich.de and login



2. Add a new or start an existing JuperLab on JUWELS vis login node or JURECA.



Xpra Integration in JupyterLab@JSC



If needed, start a new launcher by menue: File → New Launcher.
 In the launcher: click on the Xpra icon



4. Wait for the HTML desktop of Xpra. Start apps from the menue or



Xpra Integration in JupyterLab@JSC



5. Start ParaView in the Xpra environment in your browser, direct access to data stored on HPC filesystem

Use **Xpra Menue** or load modules

ml Stages/2024 GCC/12.3.0 ParaStationMPI/5.9.2-1

ml ParaView/5.12.0-RC2





How to start Xpra manually



Manual Setup of Xpra

Stop the Xpra session by xpra stop :3

with Xpra + VirtualGL

5.







Step 1: store your private ssh key in a key manager

- Windows: use pageant from the PuTTY (<u>https://www.putty.org/</u>)
- Linux: start ssg-agent (if not already running) ssh-add your private key

Step 2: login to a (visualization) login node, e.g. juwelsvis02

• Windows:

connect via a ssh client, e.g. PuTTY

Linux:

ssh <USERID>@juwelsvis02.fz-juelich.de





Step 3: load Xpra modules, check for taken display numbers and choose a free one

```
jwvis02> ml Stages/2024 GCCcore/.12.3.0 xpra/5.0.8
jwvis02> xpra list # show taken display numbers
Found the following xpra sessions:
/tmp:
```

INACCESSIBLE session at :0 INACCESSIBLE session at :1 INACCESSIBLE session at :2

Notice, that e.g. :3 is not taken. Try to start Xpra with a free display number:

```
jwvis02> xpra start :3 --start="xterm -xrm xterm*font:10x20"
...
```

Actual display used: :3

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





Step 4: check again if Xpra is running properly on this display number

| jwvis02> xpra list | |
|------------------------------------|--|
| Found the following xpra sessions: | |
| /tmp: | |
| INACCESSIBLE session at :0 | |
| INACCESSIBLE session at :1 | |
| INACCESSIBLE session at :2 | |
| LIVE session at :3 | |

• The display-number (here :3) is needed to connect to the Xpra session

Manual Setup of Xpra



Step 5: connect to Xpra session Install Xpra on your local machine. Download from <u>https://www.xpra.org/</u> (or just use your packagemanager in Linux)

Linux: use command (or GUI like Windows)

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

Windows/Linux: use Xpra GUI:



| | * | Session Launcher | - | ۵ | × |
|----------|-----------|---------------------------|----|-----|---|
| | C | Connect to xpra server | | | |
| Mode: | SSH 💌 | | | | |
| Server: | zilken1 | iuwelsvis02.fz-juelich.de | 22 | : 3 | |
| Server I | Password: | | | | |
| | | Advanced Options | | | |
| | | | | | |

Manual Setup of Xpra



Step 6: start visualization application

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

Start your application there, e.g. ParaView 5.12.0-RC2:

```
jwvis02> ml Stages/2024 GCC/12.3.0
ParaStationMPI/5.9.2-1 ParaView/5.12.0-RC2
```

Step 7: When you are done, stop the session by
jwvis02> xpra stop :3 #'xpra stop' works as well



How to start a VNC session



Remote 3D Visualization

with VNC + VirtualGL



5. start local VNC client and connect to remote display





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

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

```
ssh <USERID>@juwelsvis.fz-juelich.de
```

```
mkdir ~/.vnc
vncpasswd
```

Setup VNC Connection



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, jurecavis).
 Instead select a specific node randomly (juwelsvis00 .. juwelsvis03, jureca01 .. Jureca12)

Linux:

ssh <USERID>@juwelsvis00.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:
module --force purge
/opt/TurboVNC/bin/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 tunne

| Category: | 1.0 | | | Category: | - | | |
|---|---|---|---|---|---|---|---|
| ← Features ∧ ← Window ← Appearance ← Behaviour ← Translation ⊕ Selection | Option | ns controlling SSH | port forwarding | - Features | Option | is controlling SSH p | port forwarding |
| | Port forwarding Local ports Remote por Forwarded port | accept connection its do the same (SS is: | ns from other hosts SH-2 only) Remove | → Window → Appearance → Behaviour → Translation ⊕ - Selection | Port forwarding Local ports Remote por Forwarded port | accept connection ts do the same (SS s: | is from other hosts iH-2 only) Remo |
| Colours | | | | Colours | L5903 loc | alhost:5903 | 14 |
| Telest | Add new forwa | inded port: | 5h | Proxy | Add new forwa | rded port: | |
| Rlogin | Source port | 5903 | Add | Rlogin | Source port | 5903 | Add |
| E-SSH | Destination | localhost:590 | 3 | E SSH | Destination | localhost:5903 | 3 |
| Host keys Cipher | ● <u>L</u> ocal ● A <u>u</u> to | O Hemote | O Dynamic O IPv <u>6</u> | Host keys Cipher | Local Auto | O Remote O IPv4 | O Dynamic O IPv <u>6</u> |
| - TTY - X11 - Tunnels - Bugs - More bugs | | | | - TTY - X11 - Tunnels - Bugs - More bugs | | | |

Page 24





Step 4: start your local VNC viewer

Linux:

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

vncviewer localhost:3

Windows: Download and install turboVNC: <u>https://sourceforge.net/projects/turbovnc/</u> Connect to localhost:3

| New TurboVNC | Connection | | ? | × |
|--------------|--------------------|---------|----|------|
| TURBO VNC | C server: localhos | t:3 | • |] |
| 1.122200000 | 0 Ontinen | Connart | Ca | neel |

Start ParaView



Open a Terminal and execute:

ml Stages/2024 GCC/12.2.0 ParaStationMPI/5.9.2-1
ml ParaView/5.12.0-RC2
vglrun paraview

To stop the session:

/opt/TurboVNC/bin/vncserver -kill :3



ParaView for data visualization





- Login to jupyter-jsc.fz-juelich.de
- Start xpra or a vncserver and paraview
- Load some data, e.g.

/p/scratch/share/zilken1/HandsOn_Remote_Vis/ headsq.vti

Lets have some fun with **filters**, see next slides



Common Filters: Contour

| Properties | Information | | |
|---|---|------------|----|
| Properties | | | ₽× |
| 🕑 Арр | ly 🛛 🖉 🥘 <u>R</u> eset | 🛛 🗱 Delete | ? |
| Search (u | ise Esc to clear text) | | |
| Proper | ties (Contour 1) | | 3 |
| Contour By | Temp | | • |
| Compute Compute Compute Compute Compute | Normals Gradients Scalars Triangles 5 | | |
| Value Rang | e: [293.15, 913.15] | | |
| 1 421 | | | |
| 🕂 Display | / (GeometryRepresenta | ition) | |
| 🕂 View (F | Render View) | | |



- Extracts the points, curves, or surfaces where a scalar field is equal to a user-defined value.
- This surface is often also called an isosurface



Common Filters: Clip



Beware of data explosion:

Structured data is converted to unstructured!



- Intersects the geometry with a user-defined plane, box or sphere
- Removes all the geometry on one side of this plane (box, sphere)







Common Filters: Slice

| roperties | | |
|----------------------------|-------------------|--------------------|
| Apply C R | eset 🛛 🧱 Delet | e 🤇 🛜 |
| Search (use Esc to dear te | xt) | |
| Properties (Slice 1) | | 8 |
| Slice Type | Plane | • |
| Show Plane | | |
| Origin 0.348030680617293 | 0.596263870048398 | -0.100052940254971 |
| Normal 0.473996659300374 | -0.41018080111483 | 0.779152666278491 |
| X Normal |)[R | leset Bounds |
| Y Normal | | |
| Z Normal | | |
| Camera Normal | Cer | nter on Bounds |
| Crinkle slice | | |
| Triangulate the slice | | |
| Display (GeometryRepre | esentation) | |
| A View (Pender View) | | |



- Intersects the geometry with a plane, box, sphere or cylinder
- Similar to clipping, except that all that remains is the geometry where the plane is located.





Common Filters: Threshold

| Properties | Information | | |
|-------------|-----------------------|---------------|----|
| Properties | | | ₽× |
| Appl | y 🛛 🖉 <u>R</u> ese | t Delete | ? |
| Search (u | se Esc to clear text) | | 63 |
| Proper | ties (Threshold 1) | | |
| Scalars o | hardyglobal | | • |
| Minimum | 0 | 3.92407 | |
| Maximum | | 5.88965 | |
| All Scalars | r. | | |
| 🔲 Use Conti | nuous Cell Range | | |
| 🕂 Display | (UnstructuredGridRe | presentation) | |
| View (F | Render View) | | |



• Extracts cells that lie within a specified range of a scalar field





Common Filters: Extract Subset

| Properties | Information | | |
|-----------------------|------------------------|--------|---------|
| Properties | | | 8× |
| P ^{II} Apply | Reset | Delete | ? |
| Search (us | e Esc to dear text) | | |
| 📼 Properti | es (ExtractSubset1) | | |
| V OI | 0 | 20 | |
| | 0 | 20 | |
| | 0 | 20 | |
| Sample Rate I | 1 | | |
| Sample Rate J | 1 | | |
| Sample Rate K | 1 | | |
| Include Bou | undary | | |
| Display (| (GeometryRepresentatio | ın) | |
| 🐈 View (Re | ender View) | | |



 Extracts a subset of a grid by defining a volume of interest and a sampling rate





Exercise 2

Load

/p/scratch/share/zilken1/HandsOn_Remote_Vis/ disk_out_ref2.ex2

 Lets have some fun with filters for vector-data, see next slides

Common Filters: Glyph

| Pipeline Brow | vser | ØX |
|----------------------|---------------------------|-----|
| builti | n: | |
| 🗆 💼 disk 🛛 | out ref.ex2 | |
| Glyph | 1 | |
| | | |
| Properties | Information | |
| Properties | | 0 X |
| er Apply | 🖉 Reset 🛛 🗱 Delete 🦷 겸 | |
| Search (| use Esc to clear text) | |
| - Prope | erties (Glyph1) | 1 |
| Glyph Sou | rce | |
| Glyph Type | Arrow | |
| Orientatio | n | |
| Orientation Array | • V | |
| Scale | | |
| Scale Array | 🖉 No scale array | |
| Scale Factor | | |
| Masking | | |
| Glyph Mode | Every Nth Point | |
| Stride | 10 | |
| 💻 Displa | ay (GeometryRepresentatic | 3 |
| Representat | tion Surface | |
| Coloring | | |
| • temp | · • | |
| | 🛚 Edit 🔗 😭 😭 | |
| | | |





- Places a glyph, a simple shape, on each point (or subset) in a mesh
- glyphs may be oriented by a vector and scaled by a vector or scalar.



Page 35



Common Filters: Stream Tracer

| Properties Inf | formation | | | | | |
|-----------------------------|-------------------------|-------------------|----------------|-------------|-------------|--|
| roperties | | | | | 8 X | |
| Apply | | <u>R</u> eset | Delet | te 🛛 | ? | |
| Search (use Es | c to clear te | ext) | | | | |
| 🗖 Properties (: | StreamTrac | er 1) | | | | |
| Vectors | V | | | | • | |
| Integration Para | ameters | | | | | |
| Integration Direction | egration Direction BOTH | | | | • | |
| Integrator Type | Runge- | Runge-Kutta 4-5 💌 | | | | |
| Streamline Para | meters | | | | | |
| Maximum Streamlin Length | ie , | | 20.159 | 999847412 | 11 | |
| Seeds | | | | | | |
| Seed Type | | | Point Source | | • | |
| 🔽 Show Point | | | Ce | nter on Bou | unds | |
| Point 0 | | 0 | | 0.07999 | 99237060547 | |
| Number of Points | 100 | | | | A. | |
| Radius | 2.0159999 | 847412 | 1 | | | |
| Note: Move mou | ise and us | ie 'P' ke | ey to change p | oint positi | ion | |
| 🕂 Display (Geo | ometryRepr | esentat | ion) | | 8 | |
| 1 | | | | | | |



 Seeds a vector field with points and then traces those seed points through the (steady state) vector field.





Common Filters: Warp (vector)

| Pipeline Brow | ser | (| d X |
|---|-----------------------------|-------------|-----|
| builtin builtin disk_o bice1 WarpB | : ut_ref.ex2 yVector1 | | |
| Properties | Information | | |
| Properties | | (| ð 🗙 |
| to the second s | Ø Reset | # Delete ? | |
| Search (u | se Esc to clear i | text) | |
| - Proper | ties (WarpBy) | Ver 😰 🗊 🖸 🔒 | - |
| Vectors | • v | • | |
| Scale Factor | 0. | 1 ×. 🖸 | |
| 💻 Display | y (GeometryR | ep 📦 🗊 🖸 🔒 | |
| Representati | on Surface | - | |
| Coloring | | | |
| • temp | * | - | |
| Selit Edit | | # # 두 🔳 🖢 | |
| Styling | | | |



• Displaces each point in a mesh by a given vector field.





Calculations within ParaView

Calculator: calculates new attributes based on simple expression

- example: "LANDMASK*(abs(HGT) + 20.0)"
- Can generate vectors from scalars via "iHat*velocity_x + jHat*velocity_y + kHat*velocity_z"
- Can generate new coordinates
- Unflexibel, no "if" statement

PythonCalculator: calculates new attributes based on simple Python expression

- NumPy and SciPy functions can be used
- Can generate vectors from scalars via "make_vector (velocity_x, velocity_y, velocity_z)"

Programmable Source/Filter

- Most flexible
- Needs some deeper knowledge of ParaView conventions and data flow



| Properties | | | | | 8 |
|---------------|-----------------|----------------|--------|---------|------------|
| er Appl | ly 🦷 | 👌 Reset | 💥 Dele | ete | ? |
| Search (u | se Esc to clea | ar text) | | | 100 100 |
| 📼 Proper | ties (Calculati | or 1) | | | 6 |
| Attribute Mod | le Point [| Data | | | - |
| Result Array | Name Result | 8 | | | |
| Clear | (| | iHat | jHat | kHat |
| sin | cos | tan | abs | sqrt | + |
| asin | acos | atan | ceil | floor | - |
| sinh | cosh | tanh | х^у | exp | * |
| v1v2 | mag | norm | In | log 10 | 1 |
| V 41 V 4. | Scalars | • | | Vectors | • |
| | ocoidi o | | | | |
| Display | (GeometryR | epresentation) | | | 3 |

| Properties | Information | | | | | | |
|-----------------|--------------------|--|------------|--|--|--|--|
| Properties | | | 0 | | | | |
| PApply | @ <u>R</u> eset | * Delete | ? | | | | |
| Search (use | Esc to clear text) | | 100 | | | | |
| 😑 Propertie | es (PythonCalcul | ator4) 🗊 | 6 (C) (4)* | | | | |
| Expression | 1].PointData['R | 1].PointData['RAIN_Accumulated'], 0.0), 100.0) | | | | | |
| Array Associati | on Point Data | : Data | | | | | |
| Array Name | result | result | | | | | |
| ✓ Copy Arrays | | | | | | | |



Filter Menu:

Many more filters in the Filters Menu

| Search | Ctrl+Space |
|-------------------|------------|
| Recent | • |
| AMR | • |
| СТН | + |
| Common | • |
| Data Analysis | • |
| Material Analysis | • |
| Quadrature Points | • |
| Statistics | • |
| Temporal | • |
| Alphabetical | • |
| | |

- lists of all filters available in • ParaView (Alphabetical)
- state of the entries (enabled/disabled) depends on the current data set's type

| Histogram | | | | | | | |
|---------------------------------------|---|---------|---|----------|---|----|---|
| Integrate Variables | | | | | | | |
| Plot Data | | | | | | | |
| Plot Global Variables | Over Time | | | | | | |
| Plot On Intersection (| Curves | | | | | | |
| Plot On Sorted Lines | | | Extract Selection | | ParticlePath | | Table To Points |
| Plot Over Line | | ۲ | Extract Subset Extract Surface | | ParticleTracer Pass Arrays | | Table To Structured Grid Temporal Cache |
| Plot Selection Over T | îme , | | FFT Of Selection Over Time Feature Edges Gaussian Resampling | 2 | Plot Data Plot Global Variables Over Time Plot On Intersection Curves | | Temporal Interpolator Temporal Particles To Pathline Temporal Shift Scale |
| Probe Location | | | Generate Ids Generate Quadrature Points Generate Quadrature Scheme Dictionary | AL 10 | Plot On Sorted Lines Plot Over Line Plot Selection Over Time | | Temporal Snap-to-Time-Step Temporal Statistics Tensor Glyph |
| | Block Scalars | 0 | Generate Surface Normals Glyph | 60 | Point Data to Cell Data Principal Component Analysis Darke La anting | | Tessellate Tetrahedralize |
| P m | Cell Centers Cell Data to Point Data | | Gradient Gradient Of Unstructured DataSet | چە {} | Process Id Scalars Programmable Filter | ~ | Texture Map to Cylinder Texture Map to Plane Texture Map to Sphere |
| 3 | Clean Clean Cells to Grid Clean to Grid | © 14 | Grid Connectivity Group Datasets Histogram | | Python Annotation Python Calculator Quadric Clustering | 39 | Threshold Transform Triangle Strips |
| S S S S S S S S S S S S S S S S S S S | Clip Clip Closed Surface Clip Generic Dataset | | Image Data To AMR Image Data to Point Set Integrate Variables | | Random Attributes Random Vectors Rectilinear Data to Point Set | | Triangulate Tube Warp By Scalar |
| Re | Compute Derivatives Connectivity | | Interpolate to Quadrature Points Intersect Fragments | | Rectilinear Grid Connectivity Reflect | 2 | Warp By Vector Youngs Material Interface |
| S | Contour Contour Generic Dataset | | Iso volume K Means Level Scalars(Non-Overlapping AMR) | | Resample With Dataset Ribbon | | |
| | Convert AMR dataset to Multi-block Curvature D3 | | Level Scalars(Overlapping AMR) Linear Extrusion Loop Subdivision | | Rotational Extrusion Scatter Plot Shrink | | |
| G | Decimate Delaunay 2D Delaunay 3D | | Mask Points Material Interface Filter Median | | Slice Slice (demand-driven-composite) Slice AMR data | | |
| | Elevation Extract AMR Blocks | ~ | Mesh Quality Multicorrelative Statistics | 6 | Since Generic Dataset Smooth StreakLine | | |
| s on the | Extract Block Extract CTH Parts Extract Cells By Region | | Normal Glyphs Octree Depth Limit Octree Depth Scalars | lei | Stream Tracer For Generic Datasets Stream Tracer With Custom Source | | |
| | Extract Edges Extract Generic Dataset Surface Extract Level | | Outline Outline Corners Outline Curvilinear DataSet | | Subdivide Surface Flow Surface Vectors | | |
| | | | | | | | |

See https://www.paraview.org/Wiki/ParaView/Users_Guide/List_of_filters ٠

Calculator Extract Selection

Histogram

Animating Data



Using the Animation View, ParaView can animate

- Data time steps (if you have time-dependent data)
- Nearly any property of any pipeline object
- The camera, to perform camera flights along a specified path or orbit.
- Use Python scripts to manipulate the scene every time step

