Remote Visualization on JURECA
Remote Data Visualization on JURECA

- This talk is about **remote data visualization**
  - Data is stored on central GPFS
  - Visualization app is running on a server (JURECA)
  - Output is shown on your local desktop by means of
    - Redirection of X- and OpenGL- commands to your local X-server: *slow, maybe incompatible → bad idea*
    - Using the server components of “remote aware” visualization apps (ParaView, VisIt) on JURECA and their client components on your local machine
    - Running a virtual desktop (VNC-server) on JURECA, sending the image of this desktop to your local VNC-viewer. All vis software (client+server) is installed on JURECA: **our recommendation.**
  - In-Situ visualization is also possible (but not covered in this talk)
JURECA Visualization-Resources Hardware

12 Visualization Nodes
- 2 GPUs Nvidia Tesla K40 per node, 12 GB RAM on each card
  - 2 Login Visualization Nodes
    - jurecavis01.zam.kfa-juelich.de (jrc1383)
      jurecavis02.zam.kfa-juelich.de (jrc1384)
    - jurecavis.fz-juelich.de (jrc1383 oder jrc1384 in round-robin fashion)
  - 10 Batch Visualization Nodes
    - 8 nodes with 512 GB RAM (jrc[1385-1392])
    - 2 nodes with 1024 GB RAM (jrc139[3,4])
    - Special partition named “vis”
    - connection to vis batch nodes via login-nodes and ssh tunnels (for security reasons)

Visualization also possible on nodes without GPU’s (software rendering)
Why are Vis Nodes needed?

1. Special Software Stack on Vis Nodes:
   - **Base Software:**
     - X: X-Server, X-Client (Window-Manager)
     - OpenGL (libGL.so, libGLU.so, libglx.so), Nvidia
   - **Middleware:**
     - Virtual Network Computing: VNC-Server, VNC-Client
     - VirtualGL
     - Strudel
   - Parallel and Remote Rendering Apps, In-Situ Visualization
     - ParaView
     - VisIt

2. Usage Model:
   Vis Nodes are available for JURECA, JUQUEEN and Non-Project users
Usage Model for Vis Nodes

1. As of today:
   - JURECA projects: no problem at all, access to vis resources for all JURECA projects with the normal project contingent. BUT: use vis nodes for visualization purpose only!
   - JUQUEEN projects: if you need vis nodes, send a request to sc@fz-juelich.de, get a small contingent on JURECA for vis nodes
   - Non HPC-Project Users: if you need vis nodes, send a request to sc@fz-juelich.de, get a small contingent on JURECA for vis nodes

2. Future plans: more formal access (starting maybe next application period):
   - JURECA projects, JUQUEEN projects: request vis nodes in the application form (probably requests also possible later while project is already running). Get a small contingent for vis nodes only in addition to your normal project contingent.
   - Non HPC-Project Users: apply for test project. Get a small contingent for vis nodes only.
Entry point is

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

Docu related to VisIt:
https://trac.version.fz-juelich.de/vis/wiki/VisIt/Jureca

Docu related to ParaView:
https://trac.version.fz-juelich.de/vis/wiki/ParaView/Jureca
Remote Visualization: General Setup

User’s Workstation

- User interface display images

Firewall

Jureca
- Data access + image generation

- vis login node:
  - Direct user access
  - No accounting
  - Shared with other users
  - No parallel jobs (no srun)

- vis batch node:
  - Access via batch system
  - Accounting
  - Exclusive usage
  - Parallel jobs possible

Data

GPFS

compute node

vis login node

vis batch node
Remote Rendering with VNC/VirtualGL

Our recommendation is to use VNC for remote rendering on JURECA

- Hardware rendering (GPU acceleration) with VirtualGL
- VNC/VirtualGL is a good solution for many common OpenGL applications, e.g. IDL, Vapor, ...
- Can also be used for the frontend of “remote aware” applications like ParaView and VisIt
  - User only has to install local VNC viewer
  - Desktop sharing possible
  - Cumbersome tunneling, job submitting and starting of vnc server by hand can be avoided by using Strudel to establish the connection
Strudel (SciенTific Remote Desktop Launcher)

developed for the
- Multi-modal Australian ScienceS Imaging and Visualisation Environment
- at Monash University, Melbourne (Australia)

complex VNC scenarios become easy to use for any user
1. ssh to HPC system
2. authenticate via ssh key pair
3. submit job via slurm
4. wait for job to start (keep user informed)
5. start VNC server on node
6. establish ssh tunnel
7. start TurboVNC client and connect

complete configuration is stored in a JSON file

Download JSC version of Strudel here:

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

https://www.massive.org.au
VNC Desktop with JSC Extension ("Profiles")

JSC extension to VNC: vncserver –profile vis
Your benefits:

- nice blue JSC background 😊
- clock counting up/down
- desktop symbols for vis apps, LLview, ...
- MOTD window
- CPU, memory utilization
- GPU utilization
- VNC utilization

JSC extension
VNC extension

May 24, 2016
Scenario 1: Visualization on Vis Login Node with VNC

+ user does not need to install the vis app (ParaView, VisIt, …)
+ rendering on GPU
+ large memory
+ no batch job needed
+ no accounting
- resources shared between users
- tunnel has to be established
Scenario 1: How to Setup Visualization on Vis Login Node with VNC

1. Start vnc server on vis login node
   1.a. With Strudel (very easy)
      - [https://trac.version.fz-juelich.de/vis/wiki/vnc3d/strudel](https://trac.version.fz-juelich.de/vis/wiki/vnc3d/strudel)
   1.b. Manually
      Necessary Steps:
      - login to jurecavis
      - create vnc password (if not already done)
      - start vncserver, notice the display number
      - establish a ssh tunnel with the correct port to the vis login node
      - start local vncviewer with proper connection information

2. Start your vis app
   - load necessary modules (see documentation or use “module spider”)
   - start “vglrun paraview” or “vglrun visit –hw-accel”
Scenario 2: Visualization on Vis Batch Node with VNC

+ user does not need to install Vis-App (ParaView, VisIt, ...)
+ rendering on GPU, large memory
+ vis server can be run in parallel
  (but number of vis nodes limited to 4)
- batch job needed, accounting
- tunnel has to be established
Scenario 2: How to Setup Visualization on Vis Batch Node(s) with VNC

1. Start vnc server on vis batch node
   1.a. With Strudel (very easy)
       - [https://trac.version.fz-juelich.de/vis/wiki/vnc3d/strudel](https://trac.version.fz-juelich.de/vis/wiki/vnc3d/strudel)
   1.b. Manually
       Necessary Steps:
       - login to any JURECA login node (vis or non-vis)
       - create vnc password (if not already done)
       - generate small batch script for vnc server (if not already done)
         Note: you can allocate more the one vis node (max. 4)
       - start vncserver with “sbatch --start-x server name_of_jobscript”, notice the node name and the display number in the slurm output
       - establish a ssh tunnel with the correct node name and port
       - start local vncviewer with proper connection information

2. Start your vis app GUI (ParaView, VisIt)
   - load necessary modules (see documentation or use “module spider”)
   - start “vglrun paraview” or “vglrun visit –hw-accel”
Scenario 2: Start parallel ParaView on Vis Batch Node

Once you have established a VNC session on one (or more) vis batch nodes, you can start ParaView in parallel. Notice: all resources (nodes) are already allocated after starting the vnc server with sbatch (see step 1.a., 1.b.), so just use “srun”.

1. Start ParaView Servers:
   - open command shell, load necessary modules
   - export DISPLAY=:0.0
   - start servers e.g. by “srun --cpu_bind=none --ntasks=24 --gres=gpu:0 vglrun pvserver --use-offscreen-rendering”

2. Open ParaView GUI (load modules, start “vglrun paraview”)
3. Connect GUI to the pvserver (localhost, port 11111)
Scenario 2: Start parallel VisIt on Vis Batch Node

Once you have established a VNC session on one (or more) vis batch nodes, you can start VisIt in parallel.
Notice: all resources (nodes) are already allocated after starting the vnc server with sbatch (see step 1.a., 1.b.)

1. Open VisIt GUI (load modules, start “vglrun visit –hw-accel”)
2. Inside the VisIt GUI select the proper host profile for JURECA Vis Batch Node (documentation and download link for predefined host profiles here: https://trac.version.fz-juelich.de/vis/wiki/VisIt/Jureca)
3. Select “File open”, in the file-browser choose “JURECA Vis Batch Node” as host.
4. Select a File, choose “localhost” as launch profile, choose number of processors
Many other visualization scenarios are possible in general

Some (but not all one can think of) are covered in the documentation

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

Some examples on the next slides.....
GUI on vis login node, server on vis batch node

+ user does not need to install vis app
+ rendering on GPU
+ vis app server can be run in parallel (but number of vis nodes limited)
- batch job needed
GUI on vis login node, server on compute batch nodes

+ user does not need to install vis app
+ vis app server can be run in parallel on a really huge number of nodes
+ in situ visualization possible
- batch job needed
- only software rendering (but probably not the bottleneck)
Remote Rendering (ParaView or VisIt) without VNC

**Example: ParaView**

+ rendering on GPU
+ ParaView server can be run in parallel (but number of vis nodes limited)
  - user has to install ParaView or VisIt on his workstation
  - batch job needed
  - ssh tunnel needed

*Firewall*

ssh + tunnel
(ParaView: port 11111)