Remote Visualisation for LRZ Users
26.02.2015
Christoph Anthes
What is Remote Visualisation?

- Idea to render images based on data, which is not stored locally on the system where you are observing the rendering

- Two different approaches are common
  - Remote computation (which we will not handle in detail)
  - Remote visualisation (at which we will have a closer look)
What is Remote Visualisation?

Developed by Thomas Köckerbauer at Institute of Graphics and Parallel Processing (GUP,) JKU, Linz, AT in 2006
Remote Computation

- Remote side (server)
  - Data is stored remotely
  - Data is typically processed or generated remotely
- Transport mechanism
  - 3D content is streamed as 3D content
- Client side
  - 3D rendering is performed locally on the client side
- Implementation
  - Could be implemented by basic X-forwarding
  - Could be a proprietary solution (e.g. VisIt)
Remote Computation – Architecture

Remote Visualisation

- Remote side (server)
  - Data is stored remotely
  - 3D rendering is performed remotely
- Transport mechanism
  - Video stream is transferred
- Client side
  - Video rendering is performed locally
- Implementation
  - Remote rendering could be performed with VirtualGL
  - Local rendering could be performed with TurboVNC
What are VirtualGL and VNC

- **VirtualGL**
  - Graphics applications are either build on OpenGL or DirectX
  - With OpenGL applications it is possible to “hijack” the stream of graphics commands

- **VNC**
  - TurboVNC
  - TightVNC
  - TigerVNC
  - noVNC
Remote Visualisation – Architecture

Why is RV usable?

- Where does it make sense?
  - Improved security
  - Less complex administration
  - Reduced hardware costs
  - Reduced software costs
  - Less data transfer
  - Immediate access to visualisation of computation results

- What are the requirements?
  - Basically access to the systems
  - A client system which can somehow perform rendering
Latencies

- **Transmission delay** - The time needed to push all the bits onto the link
- **Propagation delay** - The amount of time used by a digital signal to travel from input to output
- **Queuing delay** - Time lost in routing queues, when packet is waiting to be processed

Bandwidth requirements and artefacts

- Bandwidth and compression closely related
Infrastructure – Linux Cluster

- Login nodes: lxlogin1.lrz.de to lxlogin4.lrz.de

- Render nodes: gvs1 to gvs4
  - Sun x4600 server
  - 32 cores (8 quad core opterons)
  - 256 GB main memory
  - nVidia Quadro FX 5800

- Connected directories
  - $WORK, $HOME, $SCRATCH
Infrastructure – SuperMUC RVS

- Login node(s): supermuc.lrz.de (5 login nodes)

- Render nodes: rvs1 to rvs6
  - nVidia Kepler GPGPU K20x
  - together with 128 GB RAM

- Connected directories
  - $HOME, $SCRATCH_RVS
<table>
<thead>
<tr>
<th>Software</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amira</td>
<td><a href="http://www.amiravis.com">http://www.amiravis.com</a></td>
</tr>
<tr>
<td>Paraview</td>
<td><a href="http://www.paraview.org">http://www.paraview.org</a></td>
</tr>
<tr>
<td>PyMOL</td>
<td><a href="http://pymol.org/">http://pymol.org/</a></td>
</tr>
<tr>
<td>R</td>
<td><a href="http://www.r-project.org/">http://www.r-project.org/</a></td>
</tr>
<tr>
<td>Vapor</td>
<td><a href="http://www.vapor.ucar.edu/">http://www.vapor.ucar.edu/</a></td>
</tr>
<tr>
<td>VisIt</td>
<td><a href="http://www.llnl.gov/visit/">http://www.llnl.gov/visit/</a></td>
</tr>
<tr>
<td>VMD</td>
<td><a href="http://www.ks.uiuc.edu/Research/vmd/">http://www.ks.uiuc.edu/Research/vmd/</a></td>
</tr>
</tbody>
</table>
Getting Access and Client Requirements

- Required accounts
  - Either Linux Cluster or
  - SuperMUC
  - Details were given in a previous talk

- Required SW on Linux client

- Required SW on Windows client
  - Putty ([http://www.putty.org/](http://www.putty.org/))
1. **Login to the login node**
   - Via terminal or putty

2. **Set VNC password on the render node**
   - Via terminal and ssh (potentially inside putty terminal)

3. **Reserve graphics pipe on the render node**
   - Via terminal and ssh (potentially inside putty terminal)
   - Reserve graphics pipe (via ssh on the render node)
   - Retrieve display id (will be printed in terminal)

4. **Run TurboVNC and connect** (on local machine)
   - Via terminal or click on desktop icon
   - Requires an SSH tunnel (in Win via putty)

5. **Run application** (on remote machine inside the TurboVNC window)
1. **Login to the login node**
   - `ssh supermuc.lrz.de`

2. **Set VNC password on render node**
   - `ssh rvs1.cos.lrz.de /opt/TurboVNC/bin/vncpasswd`

3. **Reserve graphics pipe on the render node**
   - `ssh rvs1.cos.lrz.de start_vnc 2`

4. **Run TurboVNC and connect**
   - `vncviewer -via <user-id>@supermuc.lrz.de`  
     `rvs5.cos.lrz.de:4` (you receive this line from 3.)

5. **Run application**
   1) `module load vmd`
   2) `vglrun vmd`
In general the same procedure as on Linux

No SSH available in command shell, thus use of putty

Logging in via SSH

Setting up a tunnel
Commands

- In general
  - `ssh` – used for connecting to remote system

- On the remote machine
  - `start_vnc` and `rvnc` – scripts running the `vncserver`
  - `vncpasswd` – setting of the VNC password
  - `module` – call to the module system loading for example prerequisites
  - `vglrun` or `rv` – accelerated call/script for application

- On the local machine
  - `vncviewer` – start the TurboVNC viewer
You can specify the duration of your VNC session
- VNC Session will automatically terminate after 12 hours
- VNC Session will automatically terminate at 4:00am
Webclient – Design

- In-house solution – idea to hide complexity and increase availability
- Designed and developed by Siew Hoon (Cerlane) Leong, VER (Distributed Resources Group)

- Support username-password (SSH) and certificate-based (GSISSH) authentication.
- Certificate-based authentication is based on MyProxy service and short lived credential service (SLCS)

- Rendering is redirected to an apache web server with a noVNC client
- User connects via web interface to web server
- Browser has to have a rather recent version
Webclient – Architecture

Requirement:
- Modern web browser

Note:
- End user device has no direct access to HPC cluster.
- Interactive access from graphics cluster to HPC cluster is disabled

Activities:
- Manage Slurm jobs and vnc server config on the graphics cluster
- Manage ssh/gsissh tunnel from the virtual machine to the graphics cluster via the HPC cluster
- Manage noVNC client
Choose your System
- Specify the duration
- Set your VNC password
- Choose your authentication method
- For example:
  - Provide your username
  - Provide your password
- Click on login
- Requirements for browser
  - Open ports: 6080-6200 (outgoing)
  - Allow pop-ups for this site
Webclient - https://vnc.drg.lrz.de/