Getting started ...

- **Tools you need**
  - SSH Clients
  - VNC Clients or WebBrowser
- **General Methodology**
  - SuperMUC Phase 2 - rstartvnc
  - More Options of rstartvnc
    - Troubleshooting
      - I don't get the vncviewer command string; instead some obscure message about not readable .err-file:
      - The -via option of my VNC client does not work!
      - I want to start an Intel MPI program, but obtain errors when trying to load the Intel MPI module in a VNC session terminal.
  - Linux Cluster - rsvnc
    - noVNC Mode
    - More Options of rsvncstart
    - Windows Users
      - Troubleshooting
        - After rsvncstart, I don't get information about the SSH connection.
        - After killing a RV job, I cannot start a new one (immediately). (Limitations)
        - The -via option of my VNC client does not work!
  - Troubleshooting not related specifically to the RV subsystems
    - VNC password (What is it? Forgotten?).
  - Advices and Hints
    - Public-Private SSH Keys and SSH Agent
    - Add SSH Tunnel to existing OpenSSH connection on command-line

Tools you need

**SSH Clients**

Under Linux and Mac OS, command-line ssh clients are mostly already installed, or installable by the package management system of the OS.

Under Windows, several SSH clients are available. Recently, also the command-line SSH tool suite is available on Windows 10 (might be installed). In the same fashion, Cygwin or MobaXterm utilize the command-line SSH tool suite. However, the most wide spread GUI based SSH tool suite for Windows is still PuTTY.

**VNC Clients or WebBrowser**

There are several VNC clients available: RealVNC, TightVNC, TigerVNC. The TigerVNC viewer in java was tested to work on several operating systems (Linux, Windows), with the nice features to have (VNC) server side rescaling when the local client window size changes. As we also use currently the TigerVNC server, the TigerVNC client is currently recommended.

Please also consider the noVNC-Mode on the Linux Cluster RVS, which does not require a VNC client, but a web-browser.

**General Methodology**

1. Login to a login node via SSH
2. Start the Remote Visualization (RV) job via respective provided tools (see below)
3. Set up a SSH Tunnel (How? see below)
4. Open **locally** on your device Viewer/Browser (depending on system and preference)
5. Once finished, kill the RV job (see below)

**SuperMUC Phase 2 - rstartvnc**

A typical session looks like this.
### local> ssh hw.supermuc.lrz.de

**hw> rstartvnc**

llsubmit: The job "srv23ib.928099" has been submitted.

Current reservations:

<table>
<thead>
<tr>
<th>Id</th>
<th>Owner</th>
<th>Submitted</th>
<th>ST</th>
<th>Pri</th>
<th>Class</th>
<th>Running On</th>
</tr>
</thead>
<tbody>
<tr>
<td>srv24ib.927933.0</td>
<td>&lt;userID&gt;</td>
<td>11/19 11:35</td>
<td>R</td>
<td>50</td>
<td>vis</td>
<td>vis01ib</td>
</tr>
</tbody>
</table>

3 job step(s) in query, 1 waiting, 0 pending, 2 running, 0 held, 0 preempted

Your VNC Desktop has been started.

Please enter the following line on your local computer:

```plaintext
vncviewer -via <userID>@hw.supermuc.lrz.de vis01:2
```

To cancel please type: `llcancel srv24ib.927933.0`

From a different terminal, start the VNC client, using your account user ID,

```plaintext
local> vncviewer -via <userID>@hw.supermuc.lrz.de vis01:2
```

First you are asked for the **SSH password** to login to hw.supermuc.lrz.de. Next, you are ask for the **VNC password**.

Once you finished your RV job, please cancel this job to release the resource, via

```plaintext
hw> llcancel <job ID>
```

where the `job ID` is written below the vncviewer command string above, after the RV job was issued. If this is missing, or you forgot this information, use `1q -c vis` to get your running job ID.

### More Options of `rstartvnc`

`rstartvnc` accepts two command-line parameters, the order of which must be obeyed! The job duration, and the resolution. For both, defaults are given - 2 hours and 1024x786. If you want a RV job with a different duration, just enter

```plaintext
hw> rstartvnc 5 1280x720
```

to get a five-hour job with a VNC screen resolution of 1280x720.

Hint: Modern VNC server and client implementations (like Tiger VNC) allow for dynamic screen resolution change - specifically the server sided resolution changing ability is rather useful.

Programs using the hardware acceleration must be started by prepending `vglrun`. For instance with `paraview`:

```plaintext
vis> module load paraview
vis> vglrun paraview
```

### Troubleshooting

#### I don't get the `vncviewer` command string; instead some obscure message about not readable .err-file:

The LoadLeveler has sometimes some latency. When the `rstartvnc` script tries to parse the job's error file to acquire the necessary information for the VNC client-server connection, and does not find it, it will crash with an error.

**Solution:** You can look for the error file yourself. You can use `1q -c vis` to see whether you RV job runs at all, and on which machine (last column), and the job ID (first column). From the job ID, you can select the error file. If it is still missing after waiting say some minutes while the job is in running state (ST: R), cancel the job, and retry. When the problem persists, please contact the Help Desk of the LRZ! Otherwise, if the error file is there, you can get the DISPLAY ID (separated via colon, usually), and start the vncviewer command with that information.

#### The `-via` option of my VNC client does not work!

Some VNC client don't know this option to secure the VNC session by a SSH tunnel.

**Solution:** You can setup the SSH tunnel by yourself. The SSH tunnel can be generated via

```plaintext
local> ssh -L <local port>:vis0X:<remote VNC port> hw.supermuc.lrz.de
```
where local port is a rather arbitrary number above 1024, for example 12345. vis0X is vis01, vis02, or vis03 - depending on where your RV job is running. The remote VNC port is the display number of the VNC server + 5900. If, as above, your job runs on vis01 with display number 2, the SSH tunnel is created via (you can also use SSH option -N and -f if it's useful for you)

```sh
local> ssh -L 12345:vis01:5902 hw.supermuc.lrz.de
```

The VNC client is then started (from a different terminal) via

```sh
local> vncviewer localhost:12345
```

Important here is that the port numbers of the vncviewer command the the SSH tunnel (local port) coincide.

I want to start an Intel MPI program, but obtain errors when trying to load the Intel MPI module in a VNC session terminal.

The default LoadLeveler RV jobs have job_type = parallel, meaning that IBM MPI is necessary for this job.

**Solution:** You can cope with that problem by editing the .vis_job.ll in your HOME directory by changing the job_type to mpich, and submit it by hand via

```sh
hw> llsubmit .vis_job.ll
```

As there is now no script parsing for and telling you the VNC client-server connection information, you have to retrieve them as described below point 1 above.

**Linux Cluster - rvsvnc**

A normal session might look like the following (lxlogin5 can be replaced by any other Linux Cluster login node).

```sh
local> ssh <userID>@lxlogin5.lrz.de
mpp2-login5> module load rvsvnc
mpp2-login5> rvsvncstart -w
[INFO] Remote Visualization resources are available!
Submitted batch job 77 on cluster rvs
[INFO] A VNC job is already running (SLURM JOB ID 77 on Cluster rvs)!
[INFO] Viewer Client mode:
  Please issue
  vncviewer -via <userID>@lxlogin5.lrz.de rvs2.cos.lrz.de:5901
  OR:
  Setup a SSH tunnel via:
  ssh -L 12345:rvs2.cos.lrz.de:5901 <userID>@lxlogin5.lrz.de
  and, open locally (on your device) a VNC viewer via
  vncviewer localhost:12345
[INFO] noVNC Webserver mode:
  Please, setup a SSH tunnel via:
  ssh -L 54321:rvs2.cos.lrz.de:6080 <userID>@lxlogin5.lrz.de
  Next, open the following url in a local web browser
  http://localhost:54321/vnc.html?resize=remote
```

There are several options possible. In any case, open a new terminal (locally). You can now choose:

1. **VNC-Mode:** Enter (of course, use your own User ID)

   ```sh
   local> vncviewer -via <userID>@lxlogin5.lrz.de rvs2.cos.lrz.de:5901
   ```

   This might not work on all systems. In this case, open a SSH tunnel directly (still locally in new terminal)

   ```sh
   local> ssh -L 12345:rvs2.cos.lrz.de:5901 <userID>@lxlogin5.lrz.de
   ```

   As this opens also a session remotely, i.e. you are logged in on the login node, you need to start another terminal locally to issue the VNC client command:

   ```sh
   local> vncviewer localhost:12345
   ```

2. **noVNC-Mode:** If no VNC client is available, or functional on your system, you can also use a web browser (Firefox/Chromium work well; see below). Open a new SSH tunnel locally

   ```sh
   local> ssh -L 54321:rvs2.cos.lrz.de:6080 <userID>@lxlogin5.lrz.de
   ```
and, afterwards, a browser with the printed URL, here as example: \texttt{http://localhost:54321/vnc.html?resize=remote}. From here, you only need to \texttt{connect}, enter the VNC password. Ready!

Instead of lxlogin5.lrz.de, any other Linux Cluster login node can be used as well!

Once you finished your RV session, just enter

\begin{verbatim}
    mpp2-login5> rsvnckill
\end{verbatim}

Your session will be killed - including all running programs - after the scheduled job time has passed. Please, keep an eye on that! After killing, it takes about \textit{half a minute} until you can start another RV session.

If you interrupted your session, or lost SSH connection, or whatever conceivable, your RV session will probably still be running. So, you can login again. And issue

\begin{verbatim}
    mpp2-login5> rsvnccheck
\end{verbatim}

in order to see the status of your job, and to \textit{retrieve the connection information} again.

\section*{noVNC Mode}

\texttt{noVNC} is based on HTML5 and serves as web socket proxy for a running VNC server. Or, more cogently expressed: You can use a browser instead of a VNC viewer. \textit{With the} \texttt{-w} \textit{option of} \texttt{rsvncstart}, a \texttt{noVNC} \texttt{web server} is started together with the \texttt{VNC server} locally on the RV machine. Once connected via browser (via SSH tunnel), you should see something like this

\begin{figure}
    \centering
    \includegraphics[width=\textwidth]{noVNC.png}
    \caption{noVNC Mode}
    \label{fig:noVNC}
\end{figure}

We tested noVNC successfully on Linux and Windows for Firefox and Chromium. Edge and Internet Explorer seem to have problems on our tested systems. If you don't need the full functionality of noVNC, and cannot change the browser available on your local system, try \texttt{vnc_lite.html} instead of \texttt{vnc.html} in the URL resulting from the RV job submission above. But easier is to download a browser like Firefox. As they are available also as portable app, you don't need administrator rights.

\section*{More Options of rsvncstart}

The module rsvnc is a fully self-documented environmental module. Use manpage like

\begin{verbatim}
    mpp2-login5> man rsvnc
\end{verbatim}

to obtain an overview of that tool suite. There are actually three tools - \texttt{rsvncstart}, \texttt{rsvnccheck}, and \texttt{rsvnckill} - which comprise the RV job handling workflow.

Each of the command-line tools has a manpage. Or, use command-line option \texttt{-h}

\begin{verbatim}
    mpp2-login5> rsvncstart -h
\end{verbatim}

in order to obtain a short help message - about the function and the command-line options available.

Currently, only \texttt{rsvncstart} - the tool to create a RV job - has several command-line options. Amongst others, the user can decide on the job duration

\begin{itemize}
    \item \texttt{--time=} \texttt{(default: 2 hours)},
    \item \texttt{--w} whether to start also a noVNC web server on the RV machine \texttt{(default: no web server)}.
\end{itemize}
**Windows Users**

SSH Clients are also available for MS Windows. If you use MobaXterm, you can essentially follow the guide above. Here, only PuTTY usage is described in the context of the usage of the LRZ remote visualization.

**Troubleshooting**

**After rsvncstart, I don’t get information about the SSH connection.**

Instead, some error message might appear about a non-running job, or unavailable information. The Slurm scheduler and the job startup mechanism might have latencies. So, the necessary connection information is then not available, yet, when the rsvncstart script parses for it. **Solution:** You can use rsvnccheck any time later to retrieve the desired information - as long as a job is running, and the job output files are still there, the information is available somewhen. Just one advice: Be patient!

**After killing a RV job, I cannot start a new one (immediately). (Limitations)**

This is also a related to the latency of the cleaning phase of Slurm, and might last for about half a minute. Furthermore, a user can start only one RV session at a time, and obtains only one node during this session. So, if there is already one job - whether running or pending or whatever - the user can’t schedule a new job, yet. **Solution:** Please, kill the old job (if not already done), and wait patiently! You can use rsvnccheck to check, whether the RV resources are again available for you.

**The -via option of my VNC client does not work!**

**Solution:** On the Linux-Cluster, the option to use a SSH tunnel is already specified. Use SSH tunnel if –via does not work!

**Troubleshooting not related specifically to the RV subsystems**

**VNC password (What is it? Forgotten?).**

The VNC server requires a password to secure the login (via vncviewer or otherwise) from access by other users (what you usually also don't want). If it is not set, yet, you are required to set one on the first usage. **Solution:** Enter a password which is convenient for you to remember, and not so easy to guess by others. **And don't use the same password as that of your LRZ account!**

If you forgot your password, you can set/change it any time by issuing vncpasswd on any login node. It has effect only on VNC servers started after the change!

**Advices and Hints**

**Public-Private SSH Keys and SSH Agent**

For convenience, we recommend the use of public-private key pairs and SSH agent (Pageant under PuTTY).
Add SSH Tunnel to existing OpenSSH connection on command-line

For advanced users: OpenSSH also admits to attach SSH tunnel on an existing SSH connection. Look for escape characters in the SSH documentation. The common way should be to issue `enter` ~C `enter`. Then a ssh> prompt should appear. Now you can simply enter, for instance, `-L12345:rvsl.cos.lrz.de:5901` to add a tunnel from your local device port 12345 to rvsl.cos.lrz.de port 5901.

Diagnosing existing tunnels is but not as simple as for PuTTY. The best chance is probably `netstat -tulpn | grep ssh`. 