Remote Login

For better support of graphical applications requiring hardware acceleration we implemented a load balanced system with a few nodes each hosting a number of powerful GPUs. The GPU nodes can be conveniently reached from within and outside the DESY network without the need for an SSH-tunnel or VPN-connection.

The entry-points are:
- `max-display.desy.de` for ssh login using the FastX2 client
- `https://max-display.desy.de:3443/` for ssh login using a(ny) webbrowser

For members of the nova project two dedicated GPU nodes have been configured. If you are not a member of the NOVA project: don't use them. The entry-points are similarly:
- `max-nova.desy.de` for ssh login using the FastX2 client
- `https://max-nova.desy.de:3443/` for ssh login using (any) webbrowser
- Have a look at the NOVA project specific FastX Instructions (thanks to Jörg Hammel).

Policies

The display nodes are primarily for graphical logins and applications requiring GPU hardware acceleration. Like regular workgroup servers, the nodes can be used:
- as job-submission host
- for short computational jobs
- code compilations
- short running applications with a low cpu and memory footprint
- graphical applications requiring GPU acceleration

The nodes are NOT intended for compute intensive jobs, in particular not for computational jobs using the GPUs. The primary task of the GPUs is rendering.

The display can be accessed directly from any location, which also means that they can be attacked from any location. We will therefore:
- deploy any relevant fix immediately.
- reboot nodes on a regular basis, typically once a week.
- reboot nodes as soon as fixes to severe exploits are released.

We aim for a 24h notification before rebooting nodes, but depending on the severity of exploits nodes also might need to rebooted on very short notice. So please:
- frequently save your work
- terminate login sessions not needed anymore
- don't use "screen" and things alike to monitor job progress: submit a batch-job.

We will take the liberty to kill compute intensive or long-running jobs.

Desktop environment

We decided to offer XFCE as the only window manager. It's a very lightweight system, starts up quickly and is very easy to use. In the fastx client (or browser) always choose XFCE (VirtualGL). It's wrapped by a small python script (gfg) which selects the least occupied GPU for your XFCE session. For minor, non-graphical task you might also choose the xterminal (but XFCE is recommended in any case). For more information about XFCE consult the xfce documentation.

For information on how to use FastX, have a look at the "How to use FastX" page.

Technology

`max-display.desy.de` is a load-balanced (poise) alias for the three nodes `max-display001-3` (likewise for `max-nova.desy.de`). Opening a connection to max-display will establish a connection with the node with the lowest CPU load, and hence presumably the most responsive node. This also guarantees that the connection will work even if one or two display-nodes are down for whatever reason, so please always use max-display and not any of the node-names.

The three display broadcast information like running sessions across all nodes in the cluster, and will establish a FastX-session with the display-node running the smallest number of sessions. That's a very simple metric, but neither CPU load nor GPU are better suited for this task, and the metric should work well as long as you adhere to the policies outlined above. FastX-server will also recognize your running sessions and allow to attach to the sessions regardless on which machine they are running.
The XFCE window manager sessions are pre-configured and actually use virtualGL (vglrun) to utilize the GPUs enabling hardware rendering of desktops and applications (see http://www.virtualgl.org/About/Introduction for a short introduction). Startup of XFCE is wrapped by a small python-script which ties the XFCE process to the GPU with the smallest number of running processes. This way newly established sessions will always end up on the least occupied GPU of the least occupied machine.