Remote Visualization Service

Luigi Calori - l.calori@cineca.it
Roberto Mucci - r.mucci@cineca.it
SCAI Department – CINECA
Objective

- Perform scientific visualization on large amounts of data produced on CINECA HPC systems
  - without moving data
  - using high performance machine
Infrastructure

• Clusters:
  • **Galileo** (2 login nodes, 4 viz/service nodes)
    o Processors: 16 cores Intel(R)
    o GPU: 2 NVIDIA Tesla K40 per node
    o RAM: 128 GB/node
  • **PICO** (big data, 2 viz nodes, 2 fat nodes)
    o Processors: 20 cores Intel(R)
    o GPU: 2 NVIDIA Tesla K40 per node
    o RAM: 128 / 512 GB/node
  • The infrastructure is **scalable**: more nodes can be added will be around 60 nodes with Nvidia K80
  • **Common home** and **scratch** filesystem mounted by login node, compute nodes and visualization nodes
  • One GPFS partition shared between Galileo login and Pico
Infrastructure

- **VNC** remote servers can be either launched as processes on login node (**ssh**) or scheduled as batch job request (**PBS**)

- **Galileo**
  - **SSH**:
    - **Login 03--06**
      - Subject to interactive cpu-time limitation (10 min)

- **PBS**
  - Visual queue, 12 hour max wall time, 32 GB max memory occupation (cgroup limited)

- **PICO**
  - **PBS**
    - Visual queue, 12 hour max wall time, no cgroups

- Number of concurrent displays a user can create is limited to 2.
Technologies

- **TurboVNC**: free remote control software package that support VirtualGL. TurboVNC performs very well on high-latency, low-bandwidth networks.

- **VirtualGL**: open source package that gives any Unix or Linux remote display software the ability to run OpenGL applications with full 3D hardware acceleration. It optimizes user experience of remote 3D applications by rendering on remote GPU while streaming only the 2D result images.

- **Remote Connection Manager (RCM)**: python cross platform application developed by Cineca that simplifies and automates the steps needed for setting up a VNC connection to the visualization nodes (job submission for VNC server start, ssh tunneling, vnc client connection) and managing it (reconnection, list, close). It makes automatic all the step needed to **create**, **connect** and **kill** remote displays.
Requirements

• To use the remote visualization service you need:
  ○ valid cluster user
  ○ username authorized to use the service (superc@cineca.it)
  ○ Internet connection
  ○ Pc with linux (most of the popular distributions), windows or mac on which install the RCM
RCM - Login

1. Check credential
2. Check update
RCM - Display info

- Create a new remote display
- Refresh list of available displays
- Connect to the remote display
- Kill the remote display (kill the job)
- Information about created displays
No display available. Press 'NEW DISPLAY' to create a new one.

NEW DISPLAY  REFRESH

Get list of available queue

Select a queue: visual
Choose display dimensions:
- Full screen
- Custom

1600x950

OK  Cancel
1. Submit a job on the chosen queue that run `vncserver` and retrieve the display number.

2. Execute `vncviewer (display number)` to connect to the remote display (SSH tunnel through login node).
Execute `vncviewer (+display number)` to connect to the remote display (SSH tunnel through login node).

Delete the job that created the vnc remote display.
Viz Software on PLX

• Paraview - http://www.paraview.org/
• Blender - http://www.blender.org/
• Tecplot - http://www.tecplot.org/
• Vapor - http://www.vaa3d.org/
• StartCCM+ (under license) - http://www.cd-adapco.com/products/star_ccm_plus/
• VMD
• Chimera
• Matlab
• RCM user documentation and download page: http://www.hpc.cineca.it/services/remote-visualisation

• Service enabling request and problem report: superc@cineca.it

• VirtualGL: http://www.virtualgl.org/

• TurboVNC: http://virtualgl.svn.sourceforge.net/viewvc/virtualgl/vnc/trunk/doc/index.html
Thank you for your attention!

"You're not allowed to use the sprinkler system to keep your audience awake."