13th Summer School on SCIENTIFIC VISUALIZATION

Remote Visualization @ CINECA: RCM Remote Connection Manager

L. Calori, R. Mucci, I. Spisso
SCAI Department, CINECA, Italy

13 June 2014, Segrate, Milano

SuperComputing Applications and Innovation Department, CINECA

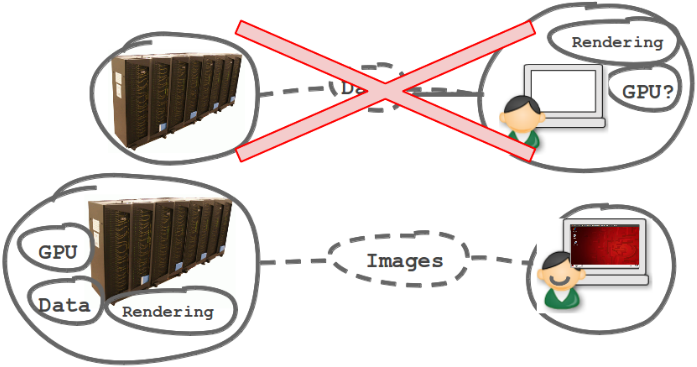
CINECA

Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

Objective of the service

- Allow users to **perform visualization** and post-processing activities **on HPC clusters equipped with GPUs** without moving datas
- **Avoiding transferring** GB of datas produced on HPC systems
- **Simplify** operations to create and manage remote displays
- Give the possibility to **share** the same display among different users



Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

Remote Visualization Infrastructure

- **PLX** cluster

  Model: IBM iDataPlex DX360M3
  Architecture: Linux Infiniband Cluster
  Number of nodes: 274 Compute + 1 Login + 6 RVN + 2 Big mem + 8 Storage
  Number of GPUs:528 nVIDIA Tesla M2070 + 20 nVIDIA Tesla M2070Q, 48 GB of RAM
  2 RNV nodes are equipped with IBM X5570 Intel(R) Xeon(R) CPU, 128 GB of RAM
  1 big memory node with a NVidia Quadro 6000 graphics card, 512 GB of RAM
- **EURORA** cluster

  Model: Eurora EUROTECH prototype
  Architecture: Linux Infiniband Cluster
  64 Compute nodes of which
  32 nodes with 2 eight-core Intel Xeon CPU with 16 GB RAM, 2 Intel Xeon Phi 5120D
  27 nodes with 2 eight-core Intel Xeon CPU with 16 GB RAM, 2 Nvidia Tesla K20s GPU
  5 nodes with 2 eight-core Intel Xeon CPU with 32 GB RAM, 2 Nvidia Tesla K20s GPU
  1024 Total cores
- **Big Data** cluster . . . coming soon

The infrastructure is **scalable**, more nodes can be added on request
Common home and **scratch** filesystem mounted by login node, compute nodes and RVN nodes

Scheduler and Queues

- **PBS Pro** batch scheduler

  version 10.4 for PLX cluster
  version 12.2 for EURORA cluster
- **Queues** available with RCM
  - **visual** queue uses a compute node with a shared memory of 47 GB on PLX and 32 GB on EURORA
  - **rvn_visual** queue uses 2 RVN nodes with 128 GB of shared memory
  - **big1_visual** queue uses the big1 node with 0.5 TB of shared memory
- **Queue policy**: by default an HPC user is enabled to use the **visual** queue, both on PLX and EURORA. The use of this queue is free accessible on HPC systems with the current billing policy.
  Upon request, it is possible to use, only on PLX, the **rvn_visual** and **big1_visual**.
- Wall Time limit of **12 hours**
- 2 concurrent displays

Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

Technologies

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

Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. **Requirements**
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

Requirements

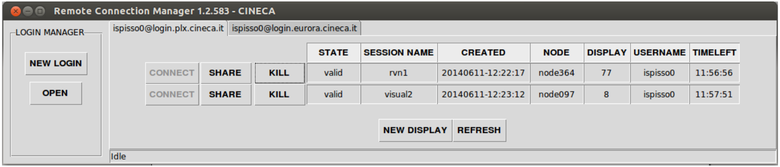
To use the remote visualization service you need:

- Valid PLX/EURORA user to use the **visual** queue
- Authorization to use **rvn_visual** or **big1_visual** queue (please write to super@cineca.it)
- Internet Connection (minimum requirement domestic ADSL)
- Pc with linux (most of the popular distributions), Windows or Mac on which install the RCM client

Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

RCM user documentation and download



Remote Connection Manager 1.2.583 - CINECA
LOGIN MANAGER
NEW LOGIN
OPEN
ispisso0@login.plx.cineca.it
ispisso0@login.eurora.cineca.it
NEW DISPLAY
REFRESH
Idle

			STATE	SESSION NAME	CREATED	NODE	DISPLAY	USERNAME	TIMELEFT
CONNECT	SHARE	KILL	valid	rvn1	20140611-12:22:17	node364	77	ispisso0	11:56:56
CONNECT	SHARE	KILL	valid	visual2	20140611-12:23:12	node097	8	ispisso0	11:57:51

- In the following slide it is shown how to download, install and use the RCM client
- The doc is available here `http://www.hpc.cineca.it/content/remote-visualization`, it lasts only 10 minutes
- Please, DO NOT do it, now. We will do a demo hereinafter

Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. **Pre-installed software**
7. Demo
8. Bonus Track: Running Paraview in client/server mode

Pre-installed software

This is a list of the pre-installed software:

The open-source softwares (no license needed to use them) are reported in **bold**

The other softwares are under license

To use them, refer to your project academic/industrial or ask the support (superc@cineca.it)

- Abaqus
- Ansys (Mechanical and Fluent)
- Comsol
- hyperstudy
- **Paraview** version 3.14, 3.98, 4.0.1, 4.1
- Pointwise version 17.0
- Tecplot version 2012R1
- **Vaa3D**
- **vmd**
- Blender (only EURORA)

. . . and much more. If your software is missing, please ask the support and, if possible, we will satisfy your request.

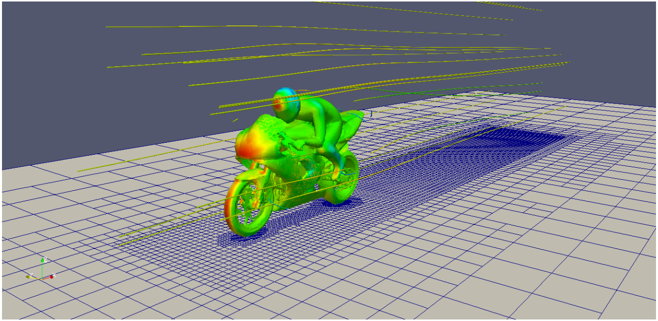
Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

Demo

Now Open the RCM client on your local workstation, and we will do two test

1. Visualization of the state of the Motorbike example in Paraview
2. Share my session by downloading motorbike.vnc available here



Outline

1. Objective of the service
2. Infrastructure
   - Remote Visualization Infrastructure
   - Scheduler and Queues
3. Technologies
4. Requirements
5. RCM: Remote Connection Manager
6. Pre-installed software
7. Demo
8. Bonus Track: Running Paraview in client/server mode

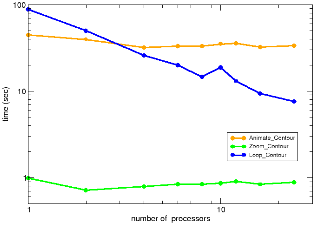
Running Paraview in client/server mode

- ParaView is designed to work in client/server mode. In this way, users can have the advantage of using a shared remote high-performance cluster to perform **parallel rendering**, as reported by Kitware
- CINECA, in the framework of a prototypical activity within PRACE, has investigated the technique of **in-situ visualization**. In-situ visualization is a term for running a solver in tandem with visualization.
- Two visualization tools that support in situ-visualization has been tested: **ParaView** and **VisIt**.
- Specifically, it has been evaluated the performance and the scalability of some rendering operations, like volume rendering, zoom of the image, contour plot, etc, as reported here

Parallel rendering benchmark I

The tests have been performed on a structured grid of 512 × 512 × 512, using

- a **single cpu**
- a **single node**, by launching the pvservers on a specific node (using a PBS job script). The tests have run on 2, 4, 6, 8, 10, 12 pvservers on the same node, using both the 2 GPUs available.
- **internode**, launching the pvserver among the computes nodes. The tests have run 16 and 24 pvserver on the 2 available nodes, using both the 2 GPUs available on each node.



Parallel rendering benchmark II

Results

- with the current benchmark data, Paraview is likely not able to exploit parallel rendering, so there is not much gain in this stage of computation
- **Pro**: only operations of contours extraction have a good scalability (and similar operations: like heavy filtering operations, or particle tracing)
- **Cons**: This client/server mode, based on parallel rendering and submission of batch job mpi, is not beneficial for this type of datas and dimensions. It needs some technical issue to be taken into account: firewall on the server, same version of the software on client and server, bandwidth usage irregular (difference between volume and surface rendering). If your datas can be loaded on a single node it is easier to use the RCM, on a standalone mode

Conclusion

- CINECA **do not** support in production this remote visualization service based on client/server mode with parallel rendering
- **RCM**, based on VirtualGL/TurboVNC, is the remote visualization production service
- If you are interested in this client/server mode, please, contact us, and we discuss your requirements to see which is the best strategy for your needs.

Questions?