Visualization support in WestGrid / Compute Canada

Alex Razoumov
alex.razoumov@westgrid.ca

copy of these slides at `http://bit.ly/wgccvis`
(will download `overview.pdf`)

Visualization support in Compute Canada

- Need to visualize output of a large numerical simulation or experimental/observed data? **We can help!**
- Our technical staff have extensive experience in scientific visualization and data analysis
  - WestGrid support support@westgrid.ca (or your local equivalent)
  - national support vis-support@computecanada.ca
  - for the past few years: "need vis. help" ✔ checkbox in RAC applications
- Front-end page `http://bit.ly/cctopviz` (team intro, visualization gallery)
- Wiki documentation (user-editable!) `https://docs.computecanada.ca/wiki/Visualization`

Visualization support in Compute Canada (continued)

- Primarily we use 3D open-source tools such as ParaView, VisIt, VTK, VMD, various Python libraries
- We can help you import your data[1] into one of these packages and write visualization scripts to automate your analysis as much as possible
  - (1) as long as its format/specification is open
  - instrument your code to output a compatible format
  - write a standalone converter
  - write a reader plugin
- Very large datasets can be visualized directly on Compute Canada systems, either interactively or via batch scripts

Yearly visualization events

- **Spring:** SEEING BIG showcase (since 2015)
  - **researchers submit visualizations to showcase their own research**
  - Mar-01 to May-31 submission window
  - entries are displayed in a video loop on a large 3840 × 2160 flat screen in the conference lobby at HPCS in June
  - submissions from a couple dozen research groups, 30-min to 60-min video
  - looking for even wider participation in 2017
- **Fall:** VISUALIZE THIS! challenge (since 2016)
  - **all participants work on the same dataset or problem**
  - competition with prizes; points awarded for interactive 3D visualization, innovative techniques to display multiple variables
  - one-month competition in 2016, likely two months in 2017
  - emphasis on creating something useful for the scientific community (techniques will be published online), suggestions welcome!
- **June:** possible visualization session at HPCS'2017
  - review submissions and techniques from most recent events
  - host a presentation from the VISUALIZE THIS! winner
  - possible visualization-themed workshop (AR/VR for scientific vis.?)

Popular open-source visualization packages

- Massive number of excellent 2D visualization tools (matplotlib, gnuplot, bokeh, various derivatives of D3.js, ...)
- Large number of 3D domain-specific (VMD, Molden, NCAR Graphics, ...) and task-specific (yt, ...) packages

We are happy to help with all of these!

... however, with large multidimensional datasets, we try to steer researchers towards scalable (parallel) 3D open-source **general-purpose scientific visualization** tools such as ParaView and VisIt

ParaView and VisIt

- Visualize any spatially-distributed data (scalar, vector, tensor fields)
- ... stored on top of any type of discretization in 2D or 3D (structured or unstructured meshes, particles, polygonal meshes, irregular topologies)
- Can handle very large datasets (GBs to TBs)
- Distributed-memory parallelism via MPI, support for parallel I/O, can scale to large ($10^3 - 10^5$ cores) computing facilities
- Interactive manipulation and support for scripting
- Understand large number (100+) of data file formats
- Provide huge array of visualization features, animation
- Open-source, multi-platform, and general-purpose
- Use OpenGL for rendering acceleration on GPUs, but can run perfectly well with software-based rasterizers and ray tracers
- Support in-situ visualization
- Built on top of VTK

VisIt 2.11

(c) 2000-2016 LLNS. All Rights Reserved.
VisIt 2.11.0, svn version 29334
August 2016

ParaView
Kitware
Sandia National Laboratories
Los Alamos NATIONAL LABORATORY EST.1943
ASC
ARL

Demo: running ParaView and VisIt locally

(1) Interactive with a couple of filters (operators)

(2) More complex visualization saved in a state (session) file

(3) Running scripts from a built-in Python shell

Remote visualization

If your dataset is on cluster.consortium.ca, you have many options:

(1) download data to your desktop and visualize it locally
limited by dataset size and your desktop's CPU+GPU/memory

(2) run ParaView/VisIt remotely on a larger machine via X11 forwarding
your desktop $\xrightarrow{\text{ssh } -\text{X}}$ larger machine running ParaView/VisIt

(3) run ParaView/VisIt remotely on a larger machine via **VNC** or **x2go**
your desktop $\xrightarrow{\text{VNC}}$ larger machine running ParaView/VisIt

- ➤ any node with X11 server; preferably a GPU compute node, could be a CPU node; scheduled or a login/development node with/without a GPU

(4) run ParaView/VisIt in **client-server mode**
PV/VisIt viewer on your desktop $\rightleftharpoons$ PV/VisIt on larger machine

(5) run ParaView/VisIt via a GUI-less batch script (interactively or scheduled) – ideal for large routine visualizations

Remote PV/VisIt via VNC on WestGrid (page 1 of 2)

full details at `http://bit.ly/remotevnc`

(1) Install TigerVNC (`http://tigervnc.org`) or TurboVNC (`http://www.turbovnc.org`) on your desktop

(2) Log in to parallel.westgrid.ca and run the command *vncpasswd*, at the prompt set a password for your VNC server (don't leave it empty) – you'll use it in step 6

(3) **Submit an interactive job** to the cluster:

qsub -q interactive -I -l nodes=1:ppn=1:gpus=1,walltime=1:00:00

When the job starts, it'll return a prompt on the assigned compute node.

(4) On the compute node **start the vncserver**:

vncserver

It'll produce something like *"New 'X' desktop is cn0553:1"*, where the syntax is *nodeName:displayNumber*

Remote PV/VisIt via VNC on WestGrid (page 2 of 2)

full details at `http://bit.ly/remotevnc`

(5) On your desktop **set up ssh forwarding** to the VNC port on the compute node:
ssh username@parallel.westgrid.ca -L xxxx:cn0553:yyyy
Here xxxx = 5901 is the local VNC port, and yyyy = 5900 (VNC's default) + *displayNumber* and is usually 5901 as well

(6) **Start VNC viewer** on your desktop, connect to *localhost:1* (that's xxxx-5900) and then enter the password from step 2 above

(7) A remote Gnome desktop will appear inside a VNC window on your desktop

(8) Inside this desktop start a terminal, use it to **start ParaView/VisIt with a VirtualGL wrapper**
vglrun /global/software/ParaView/ParaView-4.4.0-Qt4-Linux-64bit/bin/paraview
vglrun /scratch/ParaView-5.1.2-Qt4-OpenGL2-MPI-Linux-64bit/bin/paraview
vglrun /global/software/visit/visit271/bin/visit

Remote visualization in WestGrid (continued)

- Can start a VNC session on any compute node on any cluster
  - as long as this node has X11/VNC servers installed
  - let us know if you need help
- Don’t need a GPU to run ParaView/VisIt
  - a number of open-source, software-based rasterizers and ray tracers: mesa-llvm, and more recently Intel’s OSPRay and OpenSWR

... and across the country

- **Compute Ontario**
  - in **SciNet** users can start a VNC server on one of GPC's head nodes (no GPU) or on one of two interactive visualization nodes (each with two NVIDIA Tesla M2070 GPUs)
    - no scheduling
    - in either case behind the login node, so need to set up SSH port forwarding
    - run a VNC client on your laptop
  - **SHARCNET** implemented a visualization VDI server running several Linux distributions (CentOS6, Fedora20, Fedora23) via containers
    - mounting cluster filesystems via a 10 Gbit/s link
    - can log in and start a remote desktop session from your laptop's VNC client or via a web browser (HTML5-based noVNC) at `https://www.sharcnet.ca/my/systems`
    - two K1 cards (4 GPUs each); individual GPU sharing among multiple OpenGL users via VirtualGL
- **Calcul Quebec** users can connect to an x2go server on the login node of Colosse (no GPU)
  - x2go also open-source, similar to VNC, need an x2go client on your laptop
  - can pause/restart sessions

New national systems

New clusters **Cedar** (SFU) and **Graham** (Waterloo) online around Apr-01

- `https://docs.computecanada.ca/wiki/Cedar`
  27,696 CPU cores and 584 GPUs
- `https://docs.computecanada.ca/wiki/Graham`
  33,576 CPU cores and 320 GPUs

We are aiming to implement an interactive visualization setup on several nodes on these cluster, details yet to be determined

- how many nodes exactly
- whether accessible directly from outside (likely!)
- whether with GPUs
- if yes, how to share individual GPUs among multiple users

In addition, users will be able to run batch-mode (non-interactive) visualizations on regular compute (CPU and/or GPU) nodes via the job scheduler

ParaView or VisIt Full-day in-person workshops

Hands-on, lots of exercises, participants bring their own laptops

- Introduction to scientific visualization
- GUI overview
- Scientific datasets and formats, reading files
- Working with plots; visualization pipeline
- Working with filters/operators
- Quantitative analysis, data comparison
- Animation
- Scripting
- Remote visualization

✓ Since summer 2014: WestGrid taught 11 ParaView and 3 VisIt workshops at its partner institutions

✷ Coming up: VisIt workshops at SFU downtown campus (Feb-01), UofAlberta (Feb-28), UofCalgary (early March?)

✷ Let me know if you want a workshop at your location!

Online webinars

- Bimonthly during the academic year (January, March, May, September, November)
- One-hour long, very specific topics
- Past webinars are available with slides and video at `https://www.westgrid.ca/events/archive`
    - "Introduction to batch visualization"
    - "Graph visualization with Gephi"
    - "3D graphs with NetworkX, VTK, and ParaView"
    - "CPU-based rendering with OSPRay"
    - "Scripting and other advanced topics in VisIt visualization"

Future webinar ideas

We would like to get feedback from you about future topics – please tell us your preferences

- in the chat window on the right (mouse over the video), or
- quick email to **info@westgrid.ca** (will be read now), or
- via audio when we take questions at the end, or
- at any time quick email to **alex.razoumov@westgrid.ca**

Future webinar ideas (continued)

(1) Advanced topics in VisIt scripting, including callback functions

(2) Visualizing multi-resolution datasets in ParaView and/or VisIt

(3) Volumetric multi-resolution vis. with `http://yt-project.org`

- data defined on structured/unstructured meshes or particles

(4) Visualization on the new systems (once they are online)

(5) Distributed (MPI-parallel) visualization with ParaView and/or VisIt

(6) Client-server ParaView and/or VisIt

(7) Working with VTK objects in ParaView and/or VisIt

- ParaView's Calculator filter and VisIt's Expressions' Standard Editor don't let you work directly with VTK ...
- ParaView: Programmable Source and Programmable Filter
- VisIt: Python Expression Editor and Python Query Editor

Future webinar ideas (continued)

(8) Working with numpy arrays in ParaView and/or VisIt

(9) Rendering on (newer) CPUs with Intel's OpenSWR library (open-source, software-based OpenGL drop-in replacement)

(10) Writing custom reader plugins in ParaView and/or VisIt

(11) ParaViewWeb

(12) ParaView Cinema

(13) In-situ visualization in ParaView and/or VisIt

(14) Programming custom UIs in VisIt

Future webinar ideas (continued)

(15) Python's Bokeh, Seaborn, Plotly, other visualization libraries

- mostly 2D ⇒ very easy to do, but not exactly our first choice

(16) Constructive solid geometry (CSG) meshes in VisIt, or general unstructured (e.g., Voronoi) meshes in ParaView and/or VisIt

(17) Photogrammetric processing of images – tell us more about your requirements

- building polynomial texture maps from a set of images taken with varying lighting direction
- building 3D models from a set of images taken from various directions
- the catch: ideally should be open-source

(18) Visualization of point cloud data

(19) Interactive online visualizations – tell us more about your requirements

- building a research data portal?

(20) Any other special topic to your liking