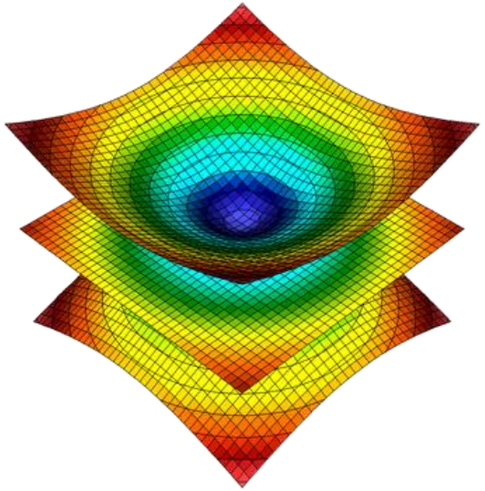


Scientific Visualization 210

ParaView and VisIt: *Scripting and Supercomputing Workflows*

KAUST Visualization Core Lab

James Kress & Thomas Theussl

Workshop Site: wiki.vis.kaust.edu.sa/training
Install VisIt 3.3.2: https://visit-dav.github.io/visit-website/releases-as-tables/#latest
Install ParaView 5.11.1: https://www.paraview.org/download/

KAUST VISUALIZATION CORE LAB

Resources

Presenter/KVL Contact Info:

- James Kress: james.kress@kaust.edu.sa
- KVL website: wiki.vis.kaust.edu.sa
- General Inquiries: help@vis.kaust.edu.sa
- KVL Vis Repo: https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes

User Resources:

- User Guide:
  - https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/
  - https://docs.paraview.org/en/latest/

Developer Resources:

- Github: https://github.com/visit-dav/visit
- GitLab: https://gitlab.kitware.com/paraview/parview

Workshop Setup

- Never logged in to Ibex before?
  - Do so now so that your scratch directory will have time to get setup
    - *ssh -X <username>@ilogin.ibex.kaust.edu.sa*
- Clone example repo on local machine
  - *git clone https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes.git*
    - ex00 - This script shows how to load a data set and then query information about the mesh, variables, and more
    - ex01 - This script shows how to create a screenshot and save it to disk
    - ex02 - This script shows how to take a series of screenshots while moving the camera and creating a movie
    - ex03 - This script shows how to animate the visualization of multiple iso surface values, showing different segments of a static data set
    - ex04 - This script shows how to animate the progress of streamlines in a flow field
    - ex05 - This script shows how to load and step through a multi time step file and take a screenshot per step
    - ex06 - This script shows the value of distributed computation, using a large data set to create a series of interesting visualizations of a cyclone simulation

Workshop Setup

mac machines

- Install ParaView
  - https://www.paraview.org/download/
  - Install XQuartz (X11)
    - https://www.xquartz.org/
- Install VisIt
  - https://visit-dav.github.io/visit-website/releases-as-tables/#latest
  - Install ffmpeg
    - https://www.ffmpeg.org/download.html
- Extra info
  - If you want to view ibex files locally without 'scp'
    - Download and install fuse and sshfs: https://osxfuse.github.io/
      - Install instructions: https://sbgrid.org/corewiki/faq-sshfs.md
  - If multiple versions of VisIt are installed, we need to add a version command to local scripts in the examples later in the workshop (e.g. "-v 3.3.2")

Workshop Setup

windows machines

- Install ParaView
  - https://www.paraview.org/download/
  - Install PuTTY
    - https://www.putty.org/
- Install VisIt
  - https://visit-dav.github.io/visit-website/releases-as-tables/#latest
  - Install ffmpeg
    - https://www.ffmpeg.org/download.html
      - Unzip this file by using any file archiver such as Winrar or 7z
      - Rename the extracted folder to ffmpeg and move it into the root of C: drive or location of your preference
      - Run the following in cmd: setx /m PATH "C:\ffmpeg\bin;%PATH%"
      - Reboot
- Extra info
  - Don't install VisIt in path with a space in it (`` '')
    - VisIt does not like this
  - I suggest running all the terminal examples in:
    - Ubuntu for Windows
      or
    - Visual Studio Code
  - If you want to view ibex files locally without 'scp'
    - Download and install SFTP Drive
      - https://www.nsoftware.com/sftp/drive/

Workshop Setup

linux machines

- Install ParaView
  - https://www.paraview.org/download/
- Install VisIt
  - https://visit-dav.github.io/visit-website/releases-as-tables/#latest
  - Install ffmpeg
    - https://www.ffmpeg.org/download.html
    - Use modules on KAUST machines
    - apt-get install ffmpeg

Workshop Notes

- Examples have been tested on the following:
  - Mac M1
    - VisIt has problems with offscreen rendering on the M1, is a future bug fix
    - ParaView has an issue with ex04, investigating
  - Mac Intel
    - Everything runs
  - Linux Ubuntu 20
    - Everything runs
  - Windows
    - Everything runs

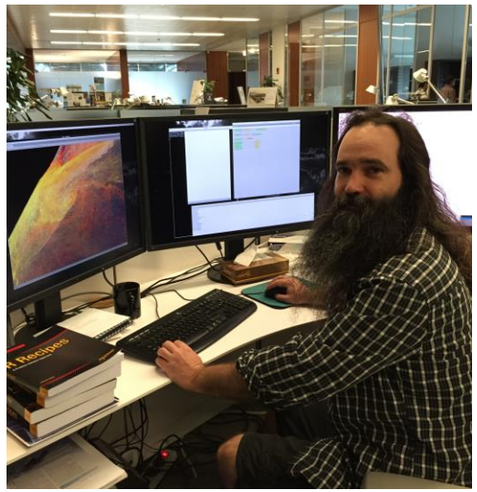
Visualization Core Lab

Overview of Facilities & Services

The Team

Dr. Sohaib Ghani
(LEAD STAFF SCIENTIST)

- VISUAL ANALYTICS
- INFORMATION VIS
- STATISTICAL ANALYSIS

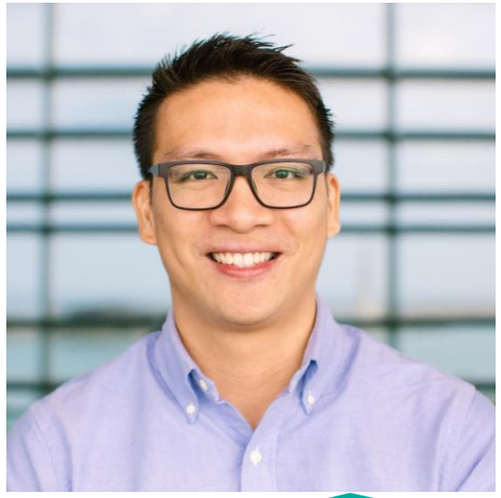


Thomas Theussl
SCIVIS

- SCIENTIFIC VISUALIZATION
- LARGE DATA ANALYSIS
- DISTRIBUTED VISUALIZATION

Dr. James Kress
HPC SCIVIS

- VISUALIZATION SOFTWARE
- HPC INSITU VISUALIZATION
- DISTRIBUTED VISUALIZATION



Dr. Ronell Sicat
VR/AR

- SCIENTIFIC VISUALIZATION
- VR DEVELOPMENT
- 3D RECONSTRUCTION

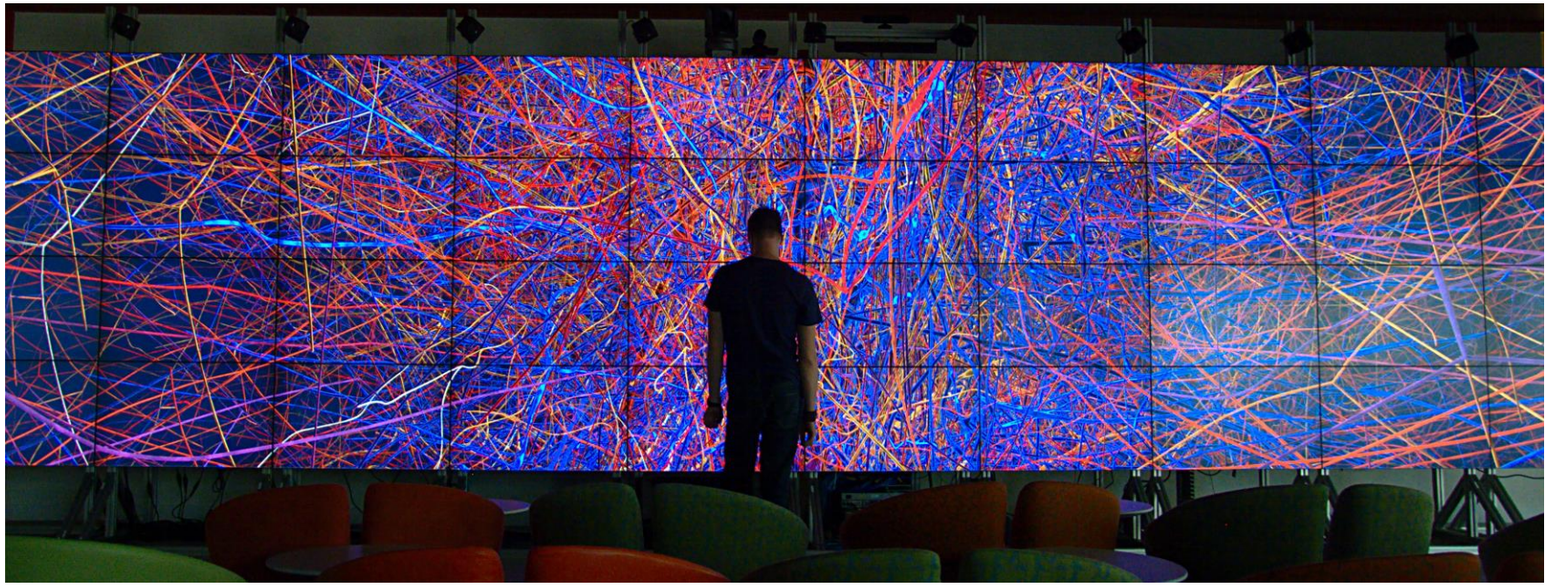
Dr. Didier Barradas
Data Scientist

- DATA SCIENCE
- MACHINE LEARNING
- DEEP LEARNING

Dr. Abdelghafour Halimi
Data Scientist

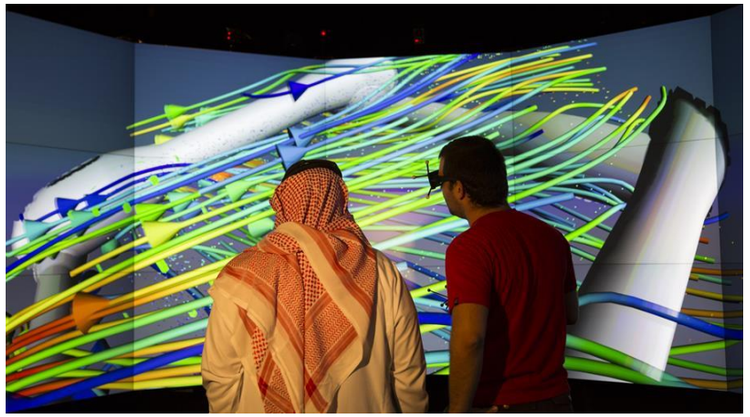
- Data Science
- Machine Learning
- Deep Learning

FACILITIES AND SPACES



ZONE 1/2 DISPLAY WALLS: 2D/3D Analytics

HMD’s



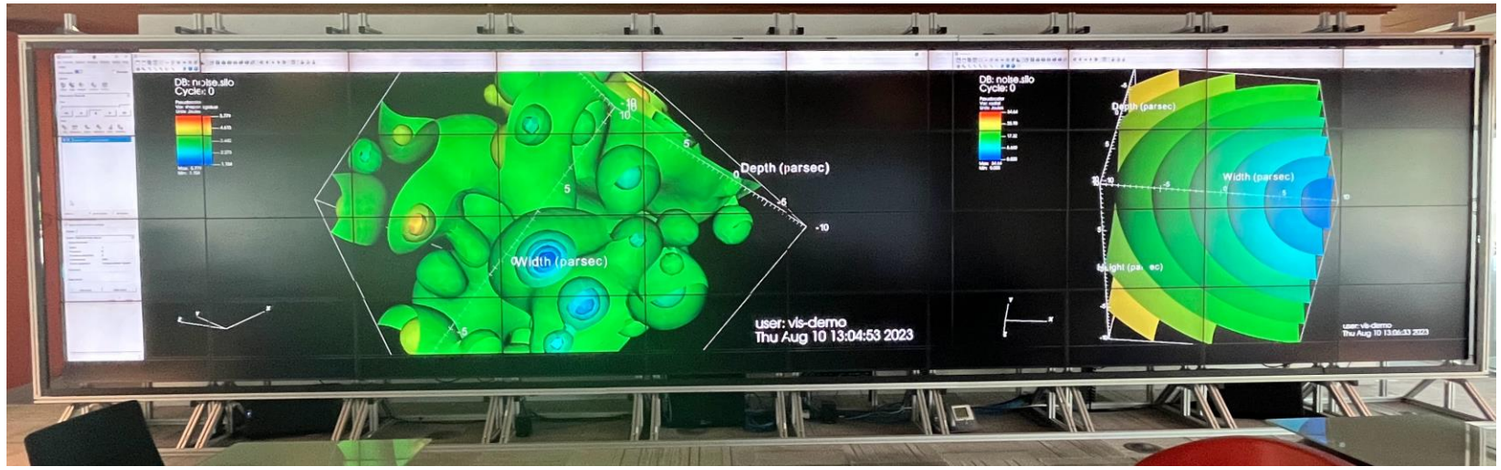
CUBES VR

ZONE 5 VR

MULTI-PURPOSE ROOM

Z2 Visualization and Collaboration

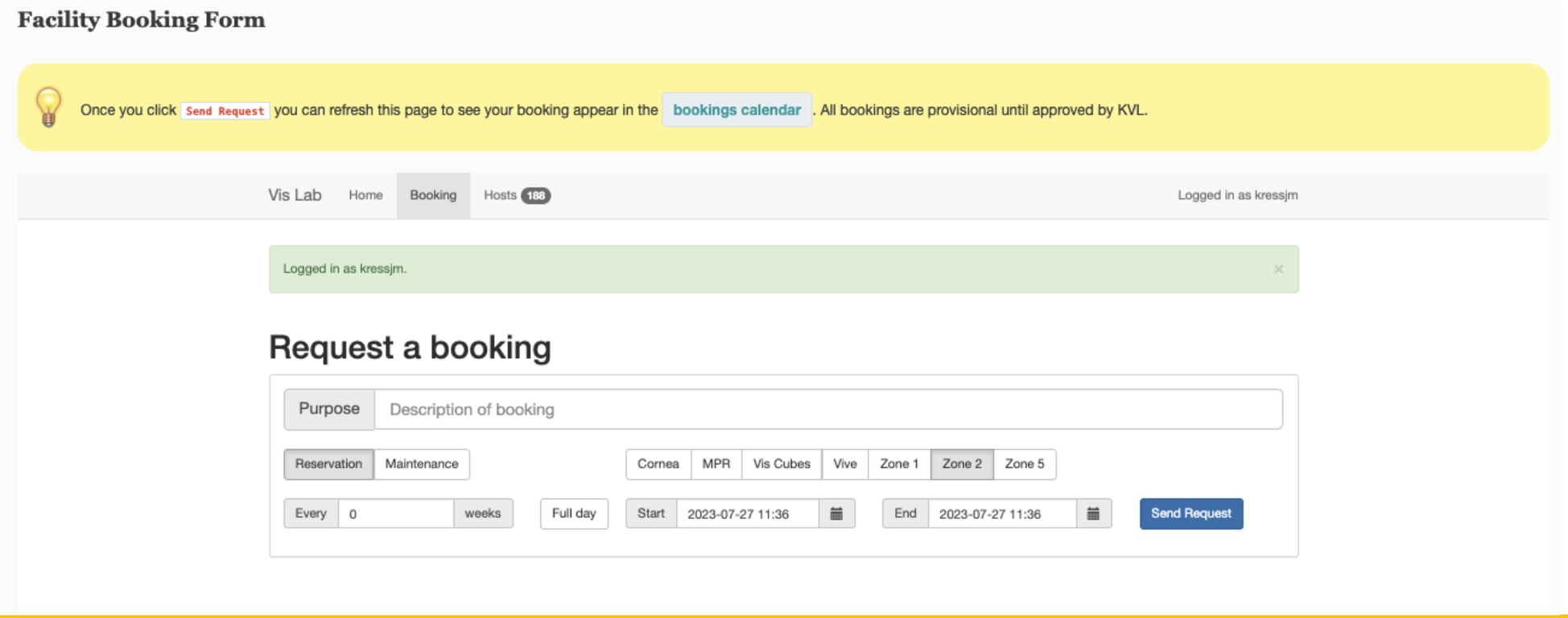
- **ParaView & VisIt on Z2**
  - Connect to Ibex for compute or other networked storage



- **Sage3 collaboration boards**
  - Software to enable teams of collaborators to work together with data in the form of data visualizations
  - https://sage3.sagecommons.org/

Accessing KVL Facilities

- Book here (requires Portal Credentials):
  - https://wiki.vis.kaust.edu.sa/booking



Collaborating with KVL

- Standard Request
  - Load data ‘X’ in program ‘P’ to produce a visualization ‘V’
- Advanced Support
  - Investigative visualization –
    - Asking “why?” & “what?” of your data
- Collaboration
  - Work with you through your research and discovery cycle

Upcoming Training Events

Scientific Visualization Workshop Series Fall 2023

Date	Training Event	Speaker	Registration
~~Sunday October 1, 2023~~	~~Scientific Visualization 101: VisIt ~ An Introductory Hands-On Workshop~~	~~James Kress~~	~~Register Now~~
Sunday October 8 2023	Scientific Visualization 210: ParaView & VisIt ~ Scripting and Supercomputing Workflows	James Kress & Thomas Theußl	Register Now
Sunday October 22, 2023	Scientific Visualization 101: Avizo/Amira ~ An Introductory Hands-On Workshop	Thomas Theußl	Register Now
Sunday October 29, 2023	Scientific Visualization 210: ParaView ~ A Plugin for Geometry Processing	Thomas Theußl	Register Now
Sunday November 5, 2023	Scientific Visualization 210: Avizo and Ilastik for Image Segmentation and 3D Analysis	Ronell Sicat	Register Now

Hands-on AI Tools and Techniques Workshop Series Fall 2023

Date	Training Event	Speaker	Registration
~~Tuesday October 3, 2023~~	~~Introduction to Machine Learning~~	~~Abdelghafour Halimi~~	~~Register Now~~
Tuesday October 10, 2023	Introduction to Deep Learning	Abdelghafour Halimi	Register Now
Tuesday October 17, 2023	Data Visualization for Data Science	Abdelghafour Halimi	Register Now

Edit

Data Science on Kaust HPC platforms Fall 2023

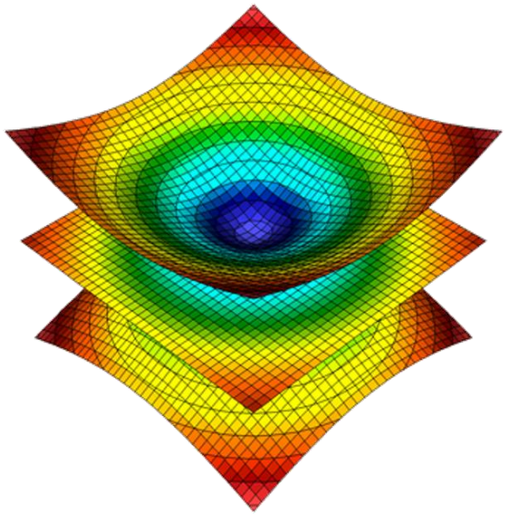
Date	Training Event	Speaker	Registration
Sunday November 20, 2023	Data Science on-boarding on KSL platforms	DB & MS	Register Now
Tuesday November 21, 2023	Distributed Deep Learning on KSL platforms	DB & MS	Register Now
Sunday November 22, 2023	High Throughput Hyperparameter Optimization of ML/DL models on KSL platforms	DB & MS	Register Now
Sunday November 23, 2023	Introduction to Containers on KSL platforms	DB & MS & AH	Register Now

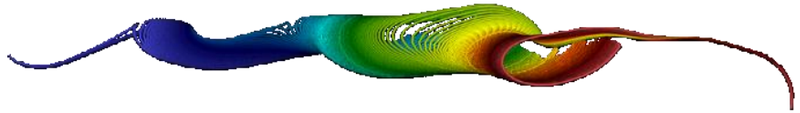
Workshop Goals

- Hands-on learning with ParaView and VisIt
  - Intermediate / advanced course
    - Scripting and workflows from desktop to HPC
  - Interactive sessions!
- Why ParaView and VisIt @ KAUST
  - Open source, scalable, multi-platform visualization applications with users worldwide
  - Available on all major computation resources at KAUST
    - Ibex and Shaheen
    - IT Remote Workstations
    - KVL Tiled-display walls

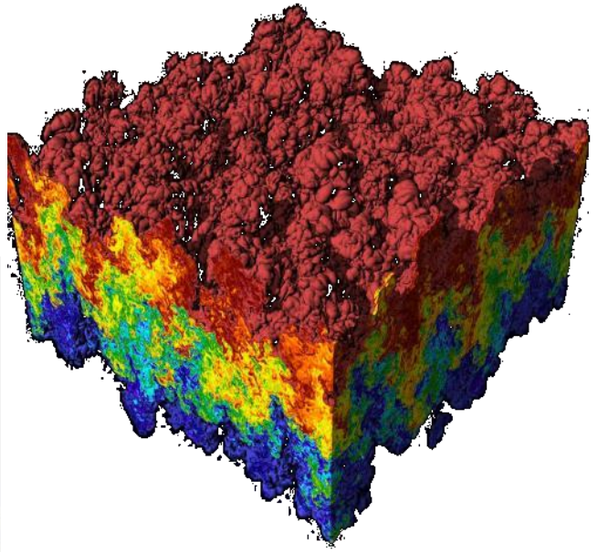
ParaView & VisIt

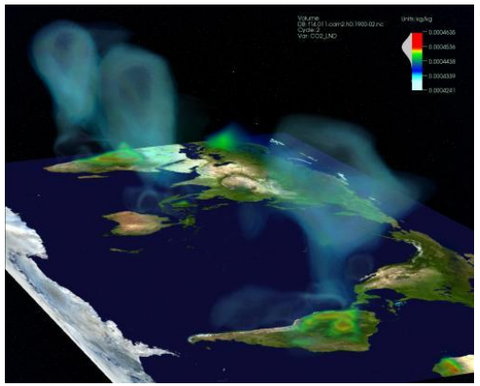
On Ibex and Shaheen



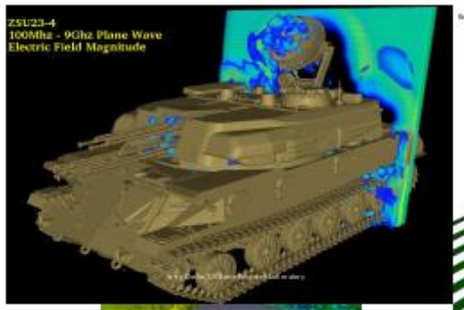


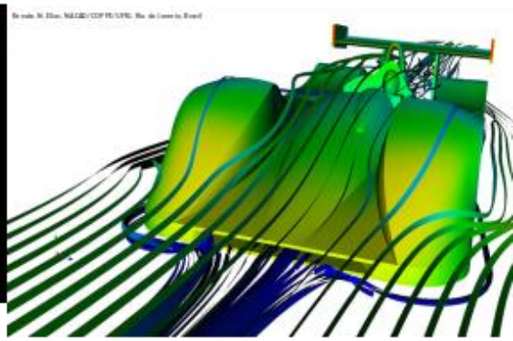
What are ParaView and VisIt?

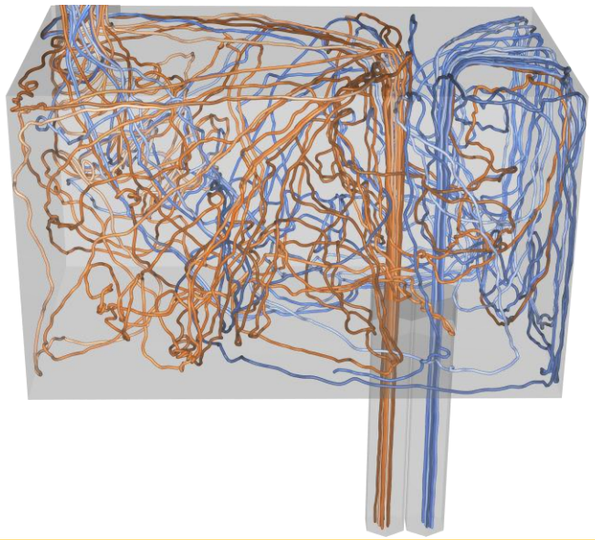


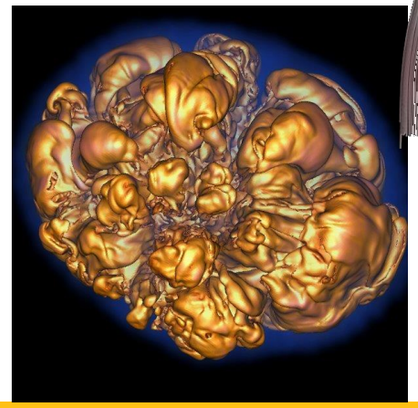


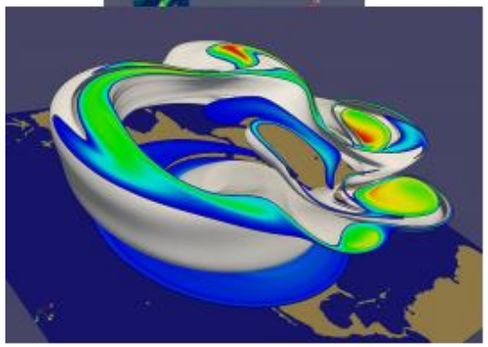
- Open-source turnkey application for data analysis and visualization of mesh-based data
- Infrastructure for parallel post-processing that scales from laptops to HPC clusters
- Built-in in situ capabilities

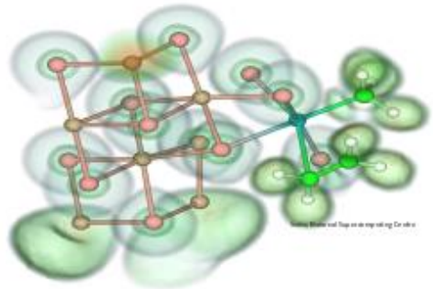












Source: Paraview Tutorial Slides

How Do I Obtain ParaView or VisIt?

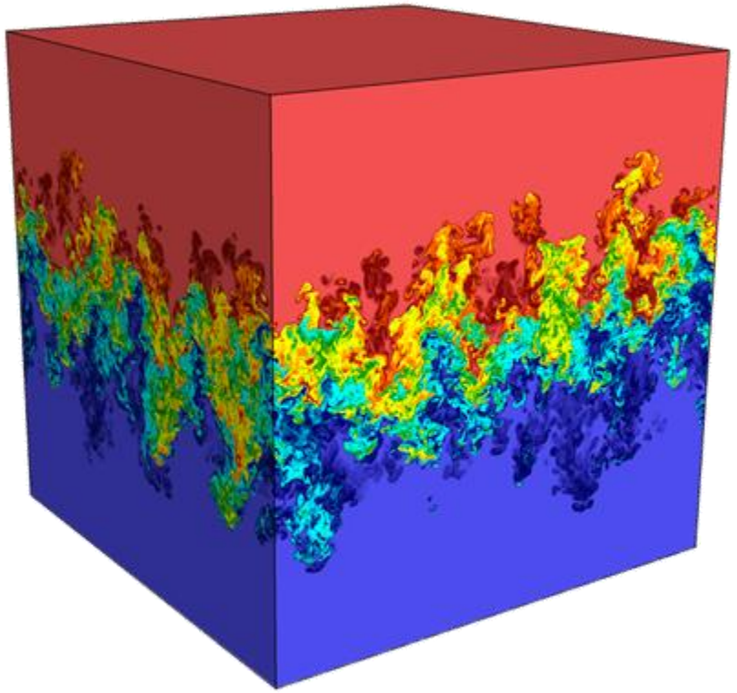
- Use an existing build:
  - For your Laptop or Workstation:
    - Binaries for Windows, OSX, and Linux (RHEL + Ubuntu)
      - https://visit-dav.github.io/visit-website/releases-as-tables/#latest
      - https://www.paraview.org/download/
  - KVL team manages builds on Ibex and Shaheen
  - IT Remote Workstations
- Build it yourself:
  - "build_visit" is a script that automates the process of building VisIt and its third-party dependencies. (docs: https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/building_visit/index.html)
  - https://gitlab.kitware.com/paraview/paraview-superbuild

Brief Look @ Core Concepts

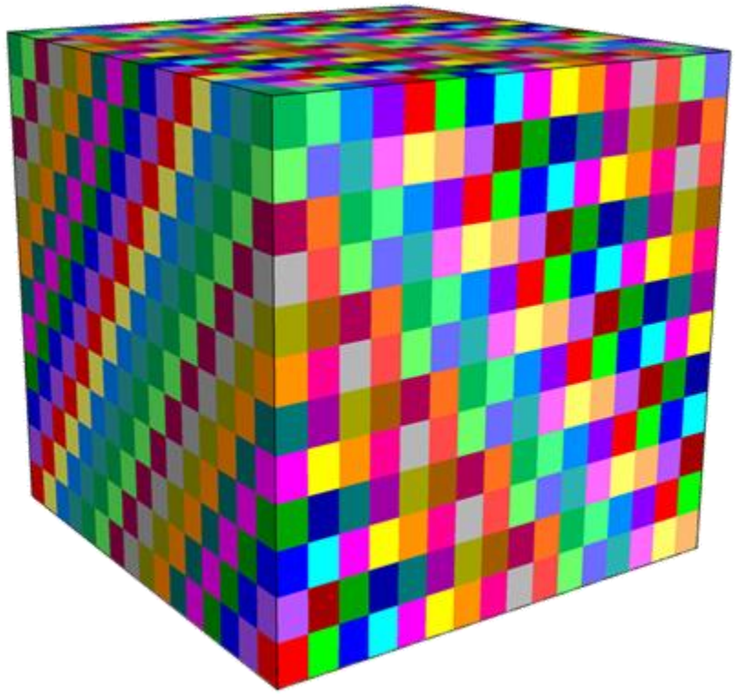
Paraview and VisIt Architecture

- **Plugin Architecture**
  - Custom File formats, Plots, Operators
  - Interface for custom GUIs in Python, C++ and Java
- **Python Interfaces**
  - Python scripting and batch processing
  - Data analysis via Python Expressions and Queries
- **In-Situ Coupling**
  - VisIt *Libsim*
  - ParaView Catalyst

ParaView and VisIt Use MPI for Distributed-Memory Parallelism

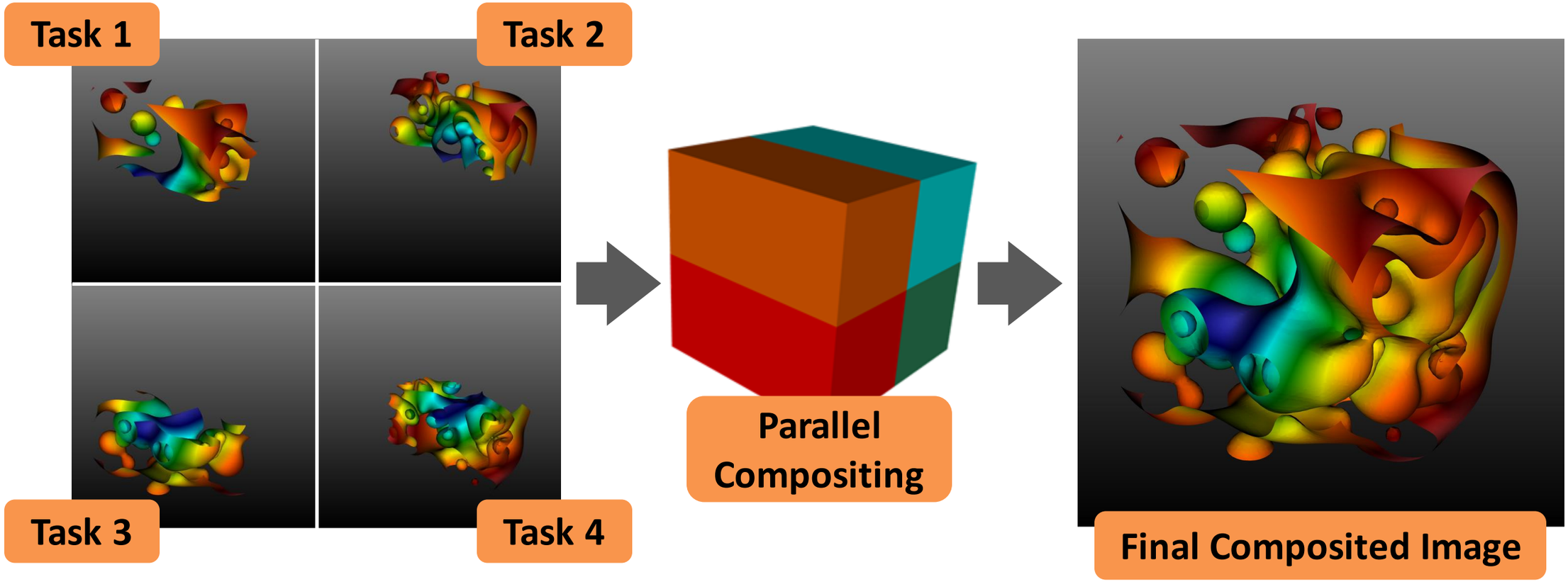


Full Dataset
(27 billion total elements)



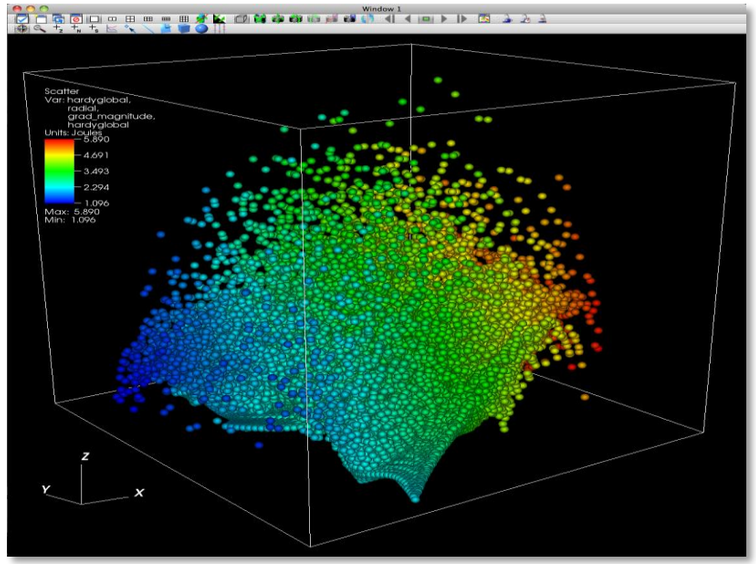
3072 sub-grids
(each 192x129x256 cells)

ParaView and VisIt Use Scalable Rendering



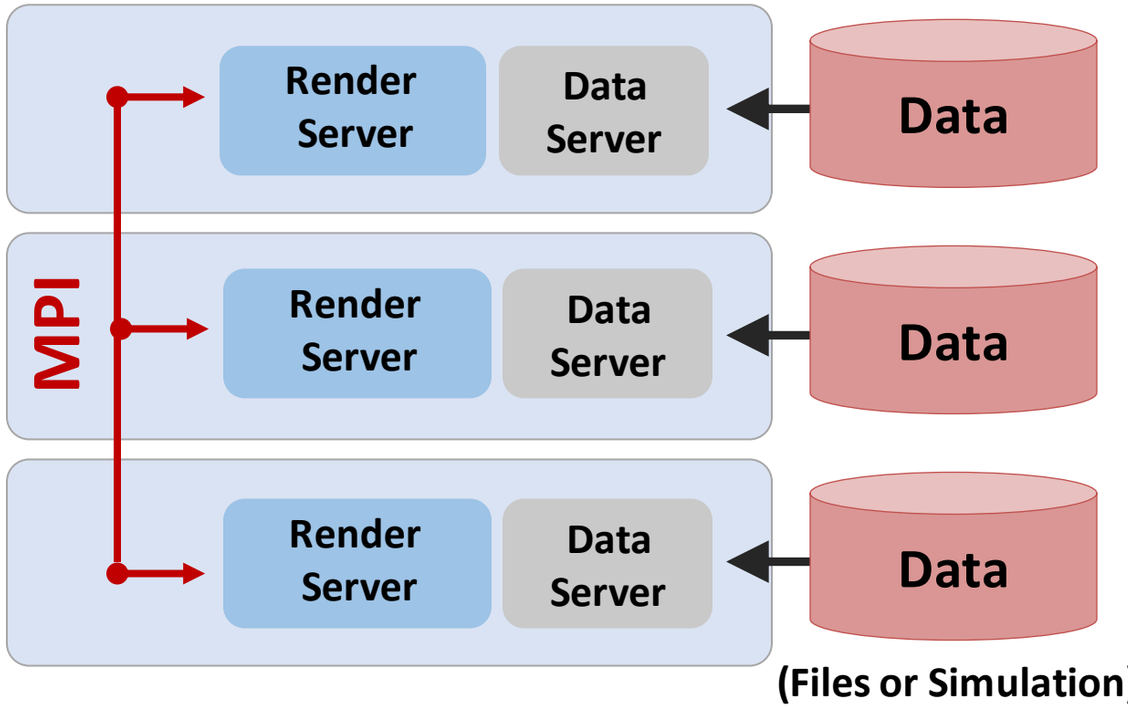
ParaView and VisIt Use a Client-Server Architecture

Client Computer



network connection

Parallel HPC Cluster



Hands–On Session 1

Ibex Interactive Visualization Overview

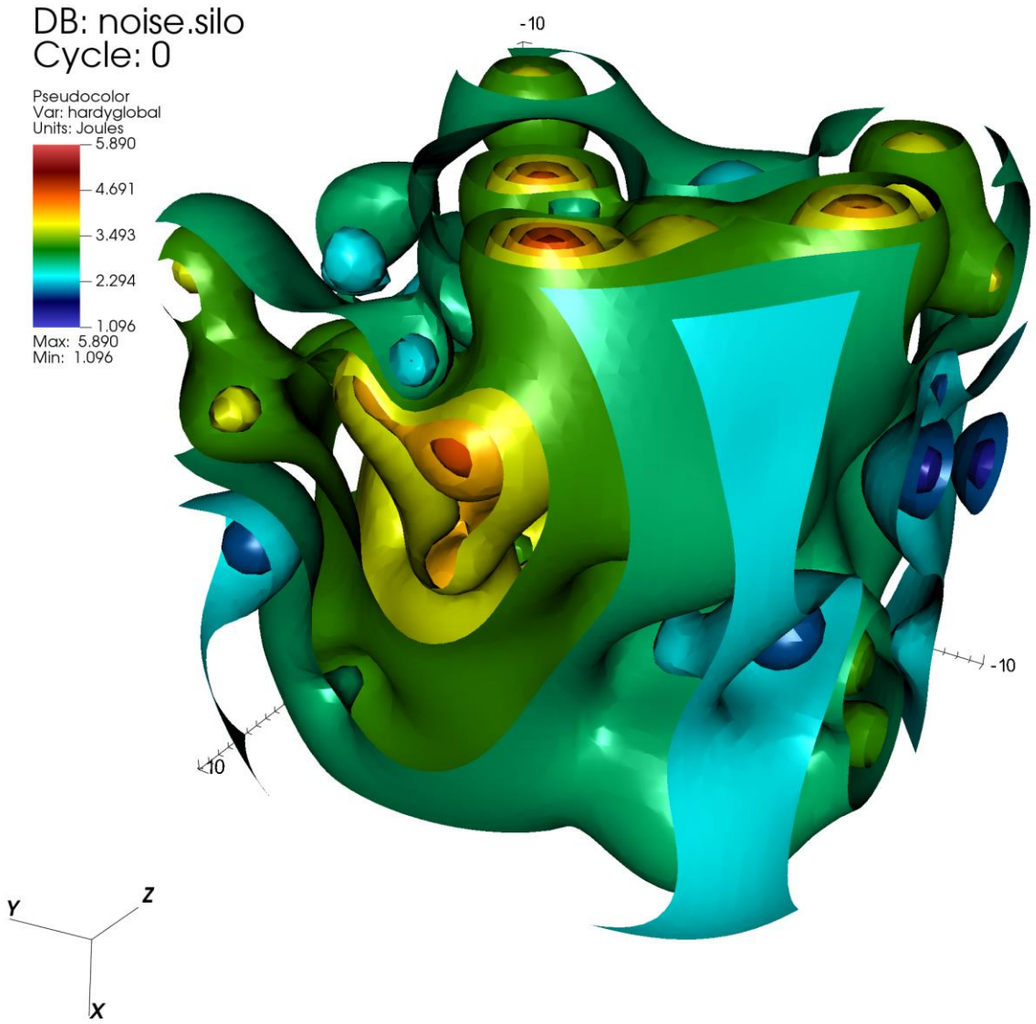
Why use ParaView and VisIt on Ibex?

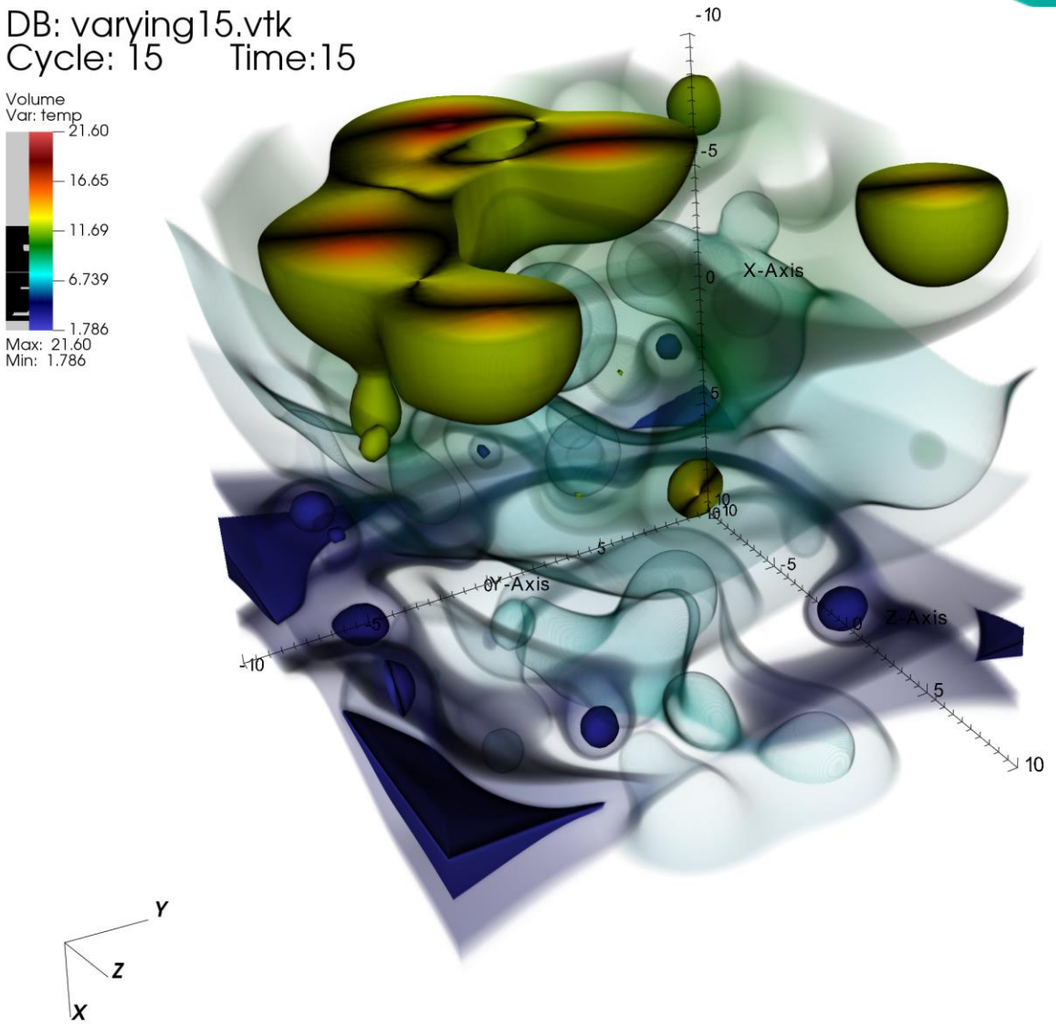
- Access to data generated on Ibex or Shaheen w/out copying
- Can use distributed computation and rendering for very large data
- Ability to run scripted batch visualization
- Ability to run client-server mode
  - GUI runs locally, all computation is done on Ibex
  - Allows for fast GUI interactions and distributed computation

Download Example Repo on Ibex

- Login to Ibex
  - *ssh <username>@ilogin.ibex.kaust.edu.sa*
- Navigate to scratch dir
  - *cd /ibex/scratch/<username>*
- Clone repo
  - *git clone https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes.git*

Repo Data


DB: noise.silo
Cycle: 0
Pseudocolor
Var: hardyglobal
Units: Joules
5.890
4.691
3.493
2.294
1.096
Max: 5.890
Min: 1.096
-10
-10
-10
Y
Z
X


DB: varying15.vtk
Cycle: 15 Time:15
Volume
Var: temp
21.60
16.65
11.69
6.739
1.786
Max: 21.60
Min: 1.786
-10
-5
0
5
10
X-Axis
Y-Axis
Z-Axis
Y
Z
X

Hands–On Session 1: Part I

Ibex Interactive Visualization with ParaView

Initially Accessing ParaView on Ibex

1. Check available ParaView versions on Ibex (use ***glogin***)

```
kressjm@login510-22:/home/kressjm$ module avail paraview

---------------------- /sw/csgv/modulefiles/applications -----------------------
paraview/5.11.1-openmpi4.0.3-egl
paraview/5.11.1-openmpi4.0.3-mesa(default)
```

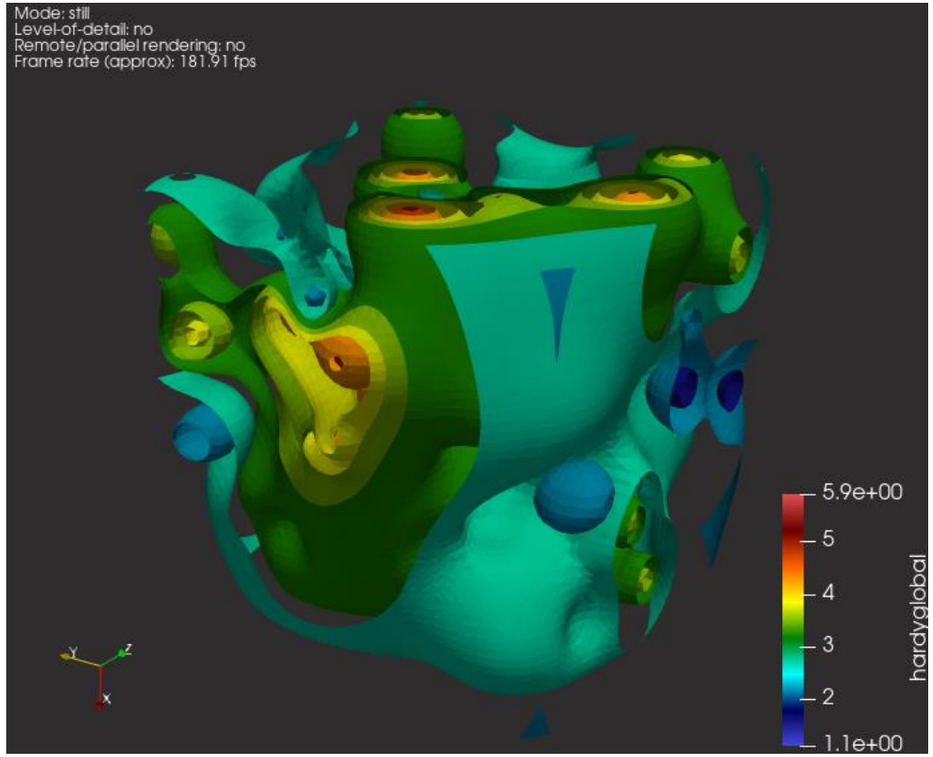
2. Download/use the latest ParaView version that ***matches*** Ibex
3. If first time using ParaView on Ibex, load the KAUST profile
   1. **Save the following servers.pvsc file to your local computer:** *ibex_server OR for Windows ibex_server_windows*
   2. **Start ParaView:** select <File/Connect> to begin
   3. **Import Servers:** Click <Load Servers> button and find the servers.pvsc file

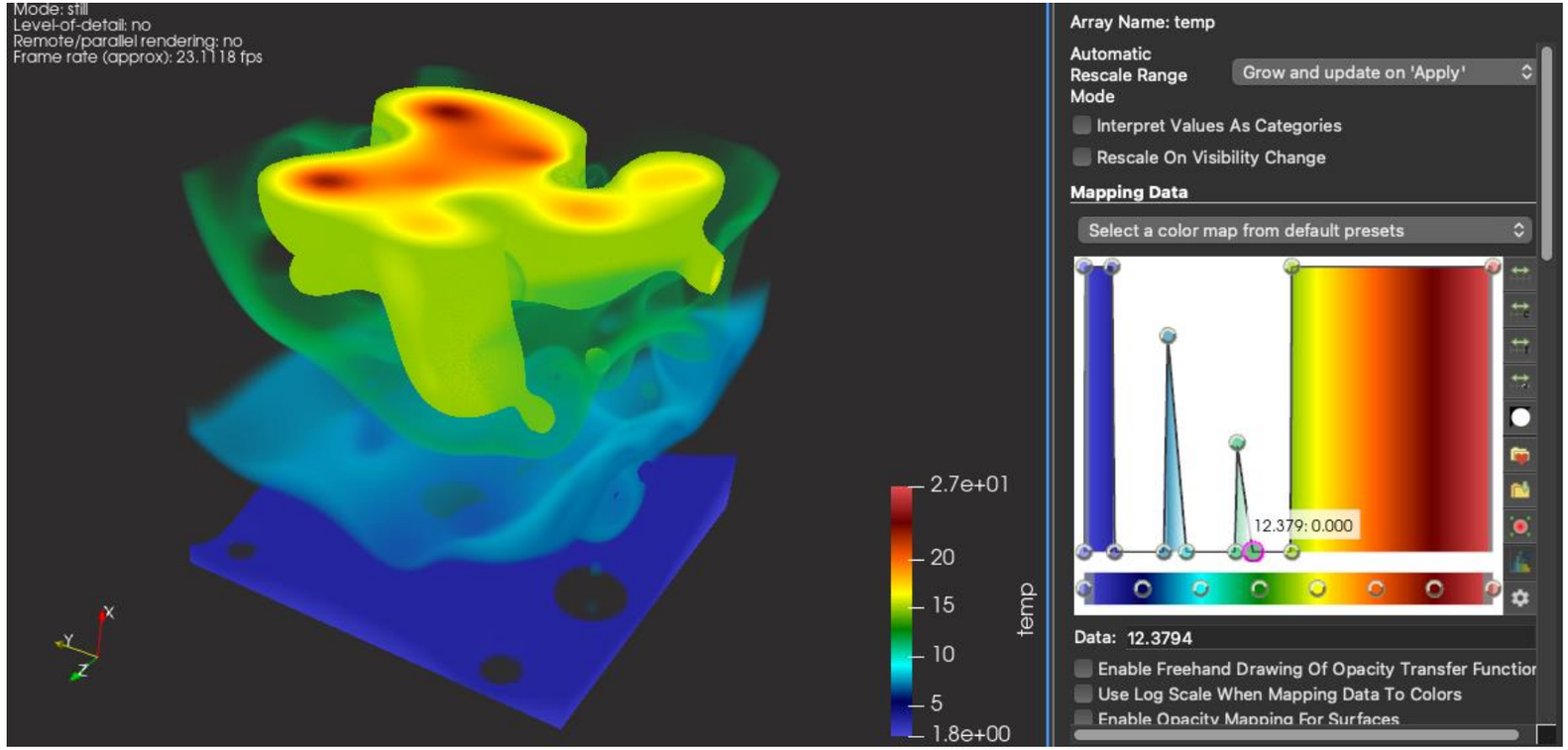
Using ParaView Interactively on Ibex

- Open **ParaView** on your local computer
- Go to: *<File/Connect>* or click the *<Connect>* button on the GUI
- Click the *<Ibex-GPU>* configuration and click connect
- Set connection options:
  - Can leave everything as default for today, just ensure *<Node Group>* is set to *"CPU"* and *<Module Variant>* is set to *"openmpi4.0.3-mesa"*
  - Click *"OK"*
- This will prompt you for your Ibex password, unless you have passwordless ssh setup
- Once specified, the server side of ParaView will be launched, and you can interact with your data (after the job launches and reaches to top of the Ibex queue)

Explore Example Repo Data Sets

- Load each of the example data sets and try different visualizations





Hands–On Session 1: Part II

Ibex Interactive Visualization with VisIt

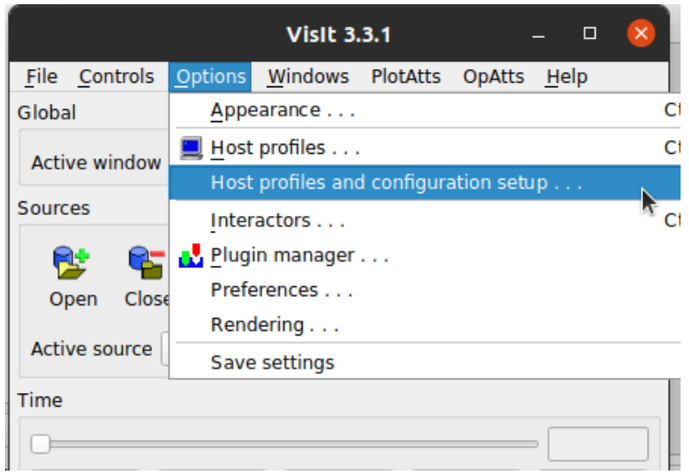
Initially Accessing VisIt on Ibex

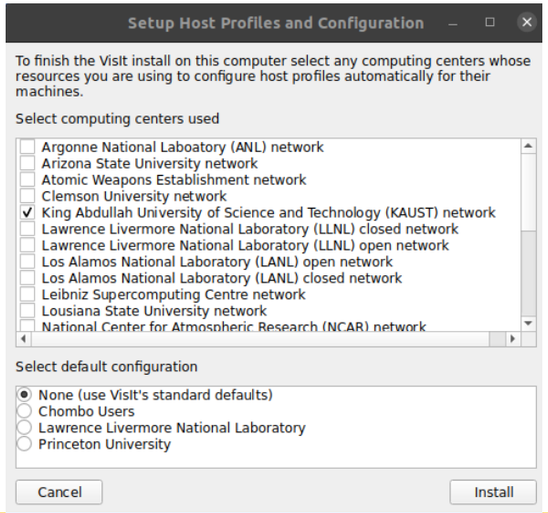
1. Check available VisIt versions on Ibex

```
kressjm@login509-02-r:/home/kressjm$ module avail visit

---------------------- /sw/csi/modulefiles/applications ----------------------
visit/3.3.2
```

2. Download/use the latest VisIt version that ***matches*** Ibex
3. If first time using VisIt on Ibex, load the KAUST profiles
    1. Click *<Options/Host profiles and …>*
    2. Select KAUST network
    3. Click *<Install>*
    4. Save settings *<Options/Save Settings>*
    5. Relaunch VisIt

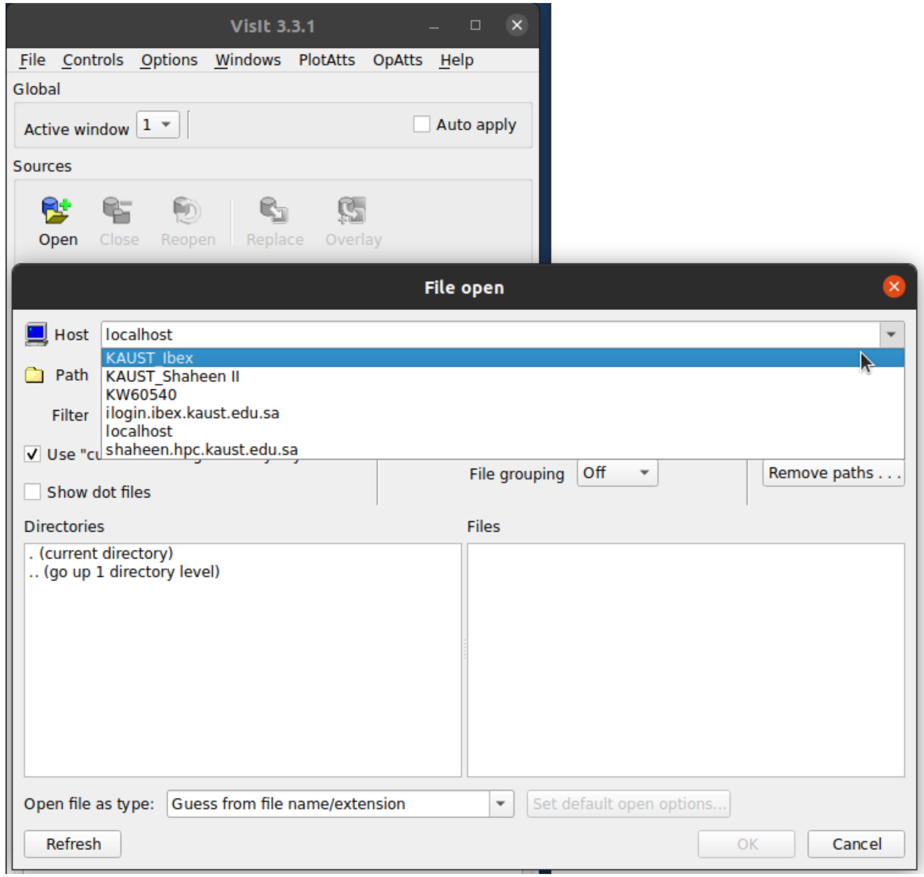




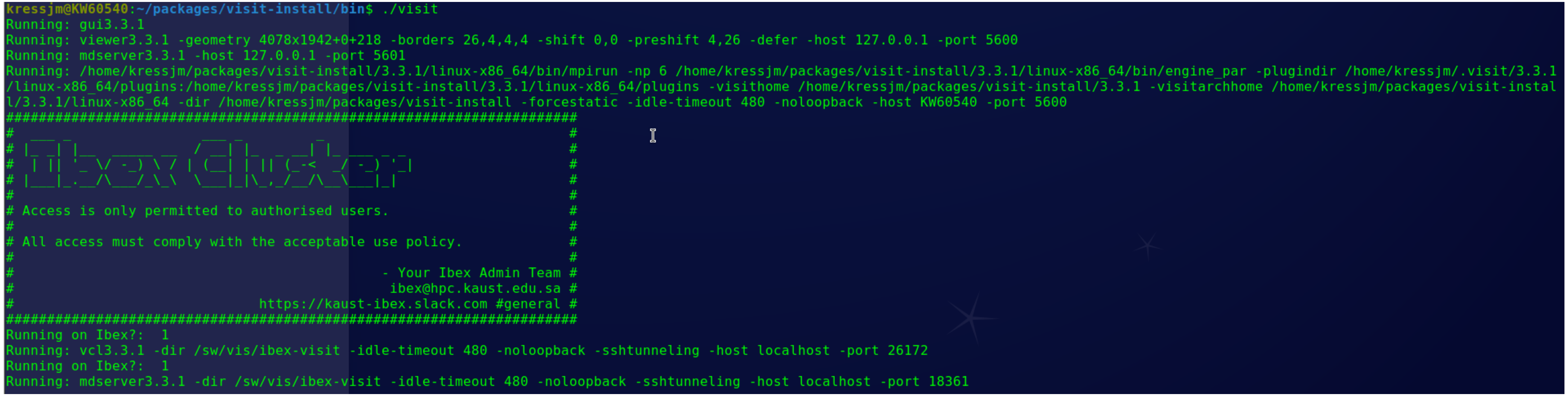
Using VisIt Interactively on Ibex

- Open **VisIt** on your local computer
- Go to: *<File/Open file>* or click the *<Open>* button on the GUI
- Click the *<Host>* dropdown menu on the *"File open"* window that popped up and choose *"Ibex"*
- This will prompt you for your Ibex password, unless you have passwordless ssh setup
- Navigate to the file you want to process
- Once you choose a file, you will be prompted for the number of nodes and processors you would like to use ***(for now, use 2 processes and 1 node)***
- Once specified, the server side of VisIt will be launched, and you can interact with your data (after the job launches and reaches to top of the Ibex queue)

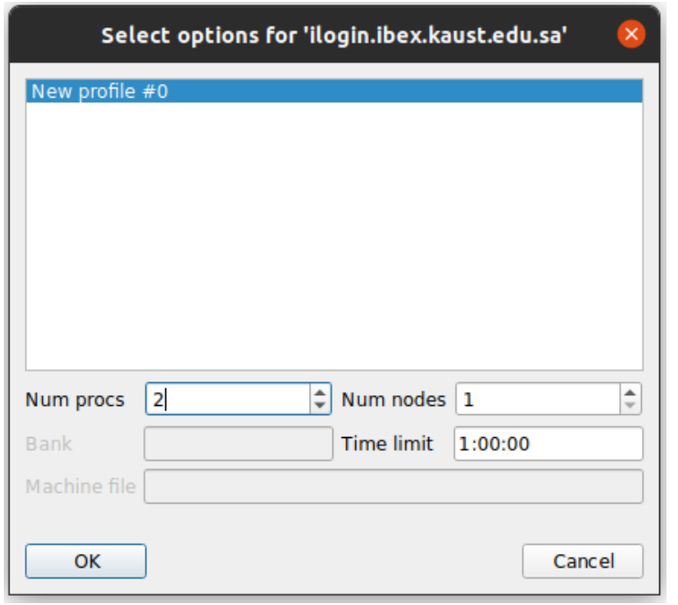
Using VisIt Interactively on Ibex Cont.



vcl – *VisIt Component Launcher (manages VisIt session)*
mdserver – *VisIt metadata server (interacts with GUI and databases)*



Using VisIt Interactively on Ibex Cont.



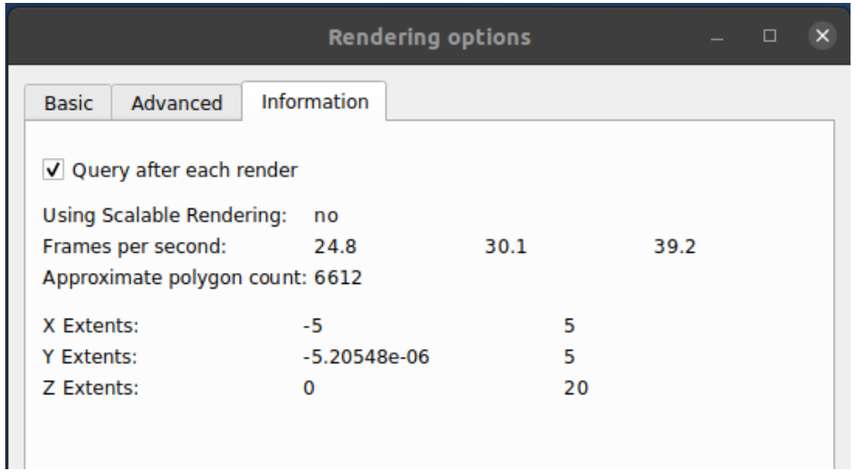

```
Running: /opt/slurm/cluster/ibex/install/bin/sbatch --export=HOME=/home/kressjm,LIBPATH=/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64/lib,LD_LIBRARY_PATH=/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64/lib/osmesa:/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64/lib/mesagl:/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64/lib:/sw/csi/gcc/8.2.0/el7.5_binary/lib64:/sw/vis/ibex-visit/bin/../3.3.1/linux-x86_64/lib,VISITHOME=/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1,VISITARCHHOME=/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64,VISITPLUGINDIR=/home/kressjm/.visit/3.3.1/linux-x86_64/plugins:/sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64/plugins --partition=batch --time=1:00:00 --ntasks=2 --nodes=1 --tasks-per-node=2 /ibex/scratch/kressjm/visit.kaust.09:00:40
Submitted batch job 23496397
```

```
kressjm@login509-02-r:/ibex/scratch/kressjm$ cat visit.kaust.09\:00\:40
#!/bin/sh
cd /ibex/scratch/kressjm
ulimit -c 0
# Submitted on host login509-02-r
echo "LIBPATH=$LIBPATH"
echo "LD_LIBRARY_PATH=$LD_LIBRARY_PATH"
echo "VISITHOME=$VISITHOME"
echo "VISITARCHHOME=$VISITARCHHOME"
echo "VISITPLUGINDIR=$VISITPLUGINDIR"
srun --export=ALL --ntasks=2 --ntasks-per-node=2 /sw/vis/ibex-gpu.2022.02/visit-src/install/3.3.1/linux-x86_64/bin/engine_par -dir /sw/vis/ibex-visit -forcestatic -idle-timeout 480 -noloopback -sshtunneling -host login509-02-r -port 15602 -key 295fbdc83b814c55d533
```

engine_par– *VisIt parallel computation engine*

Explore Example Repo Data Sets

- Load each of the example data sets and try different visualizations
- Note on rendering
  - VisIt has two rendering modes
    - Transfer data to client for rendering
      - Done when data is small
    - Transfer images to client, rendering on the server
      - This is how VisIt can render extremely large data on clusters
      - This is called scalable rendering
      - You can turn on/off scalable rendering, see stats, and other options @ *<Options/Rendering>*



Hands–On Session 2: Part I

Scripting Visualization within ParaView

Download Example Repo Locally

- Navigate to your preferred dir
- Clone repo
  - `git clone https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes.git`

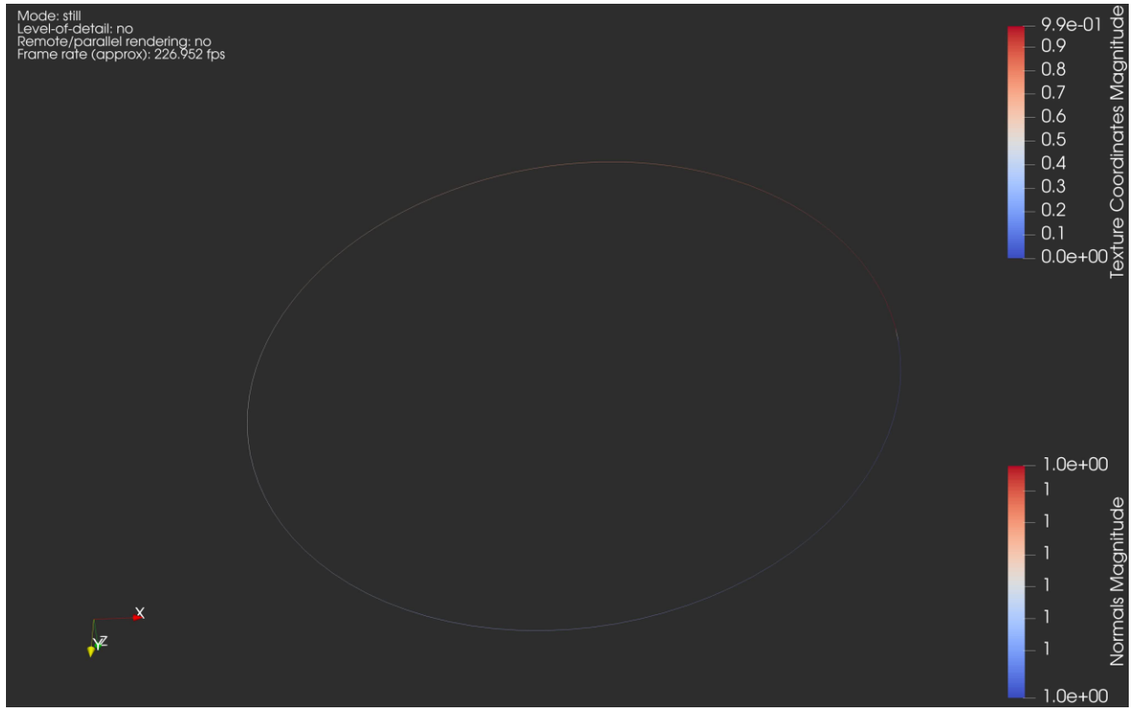
Using ParaView GUI and Python Simultaneously

- Open ParaView
  - Open Python shell: *<View/Python Shell>*

- *Go to ParaView Docs*
  - https://docs.paraview.org/en/latest/Tutorials/ClassroomTutorials/pythonAndBatchParaViewAndPython.html
  - We'll walk through some of the initial copy-paste examples

Running a Script Interactively in ParaView

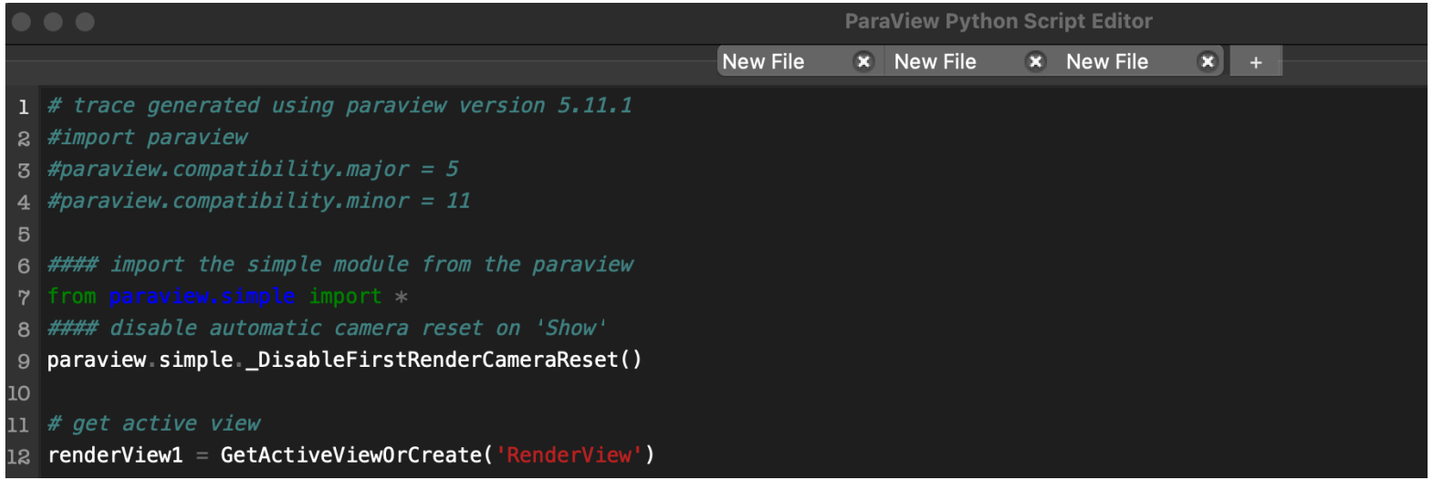
- Works just like the previous examples, but code is in a file
  - Use your favorite editor
  - Have more complicated multi-file scripts

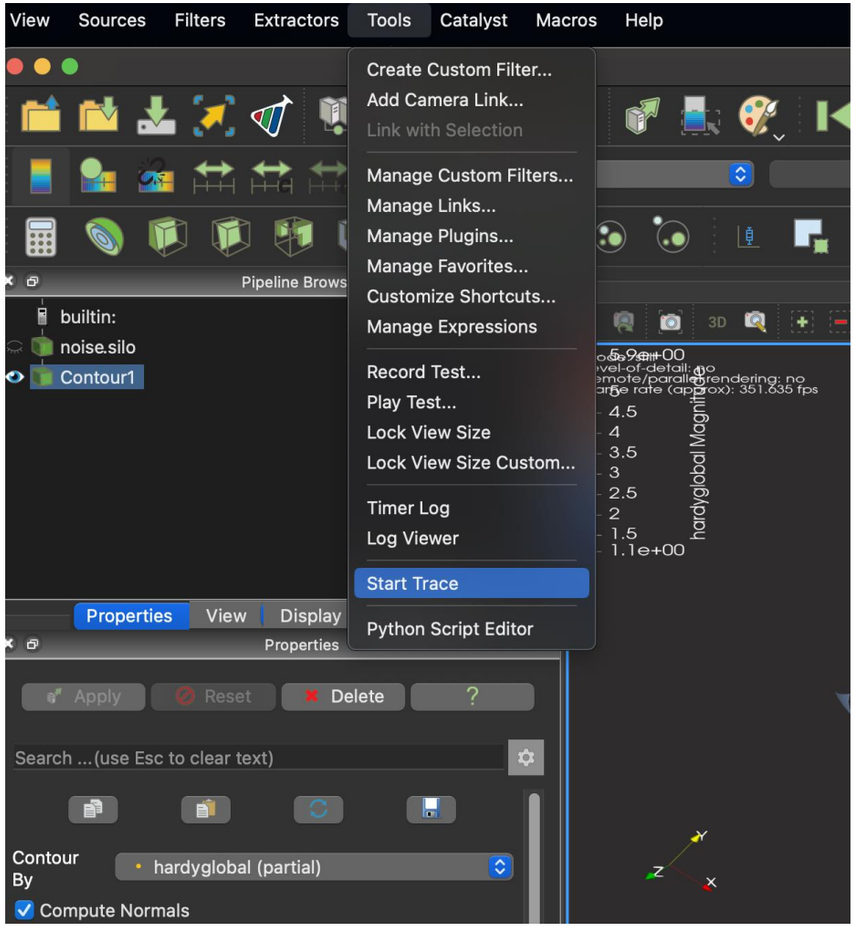


- Open the Python Shell
  - Click *<Run Script>*
  - Locate and run
    "/path/to/KAUST_Visualization_Vignettes/ParaView_Vignettes/ex02_pvAnimation/ex02_pvAnimation.py"

Tracing Your Actions

- Open ParaView
  - Start Trace: *<Tools/Start Trace>*
  - Interact with the GUI to do the vis you want
  - Stop Trace: *<Tools/Stop Trace>*
  - A lengthy trace will reproduce your actions



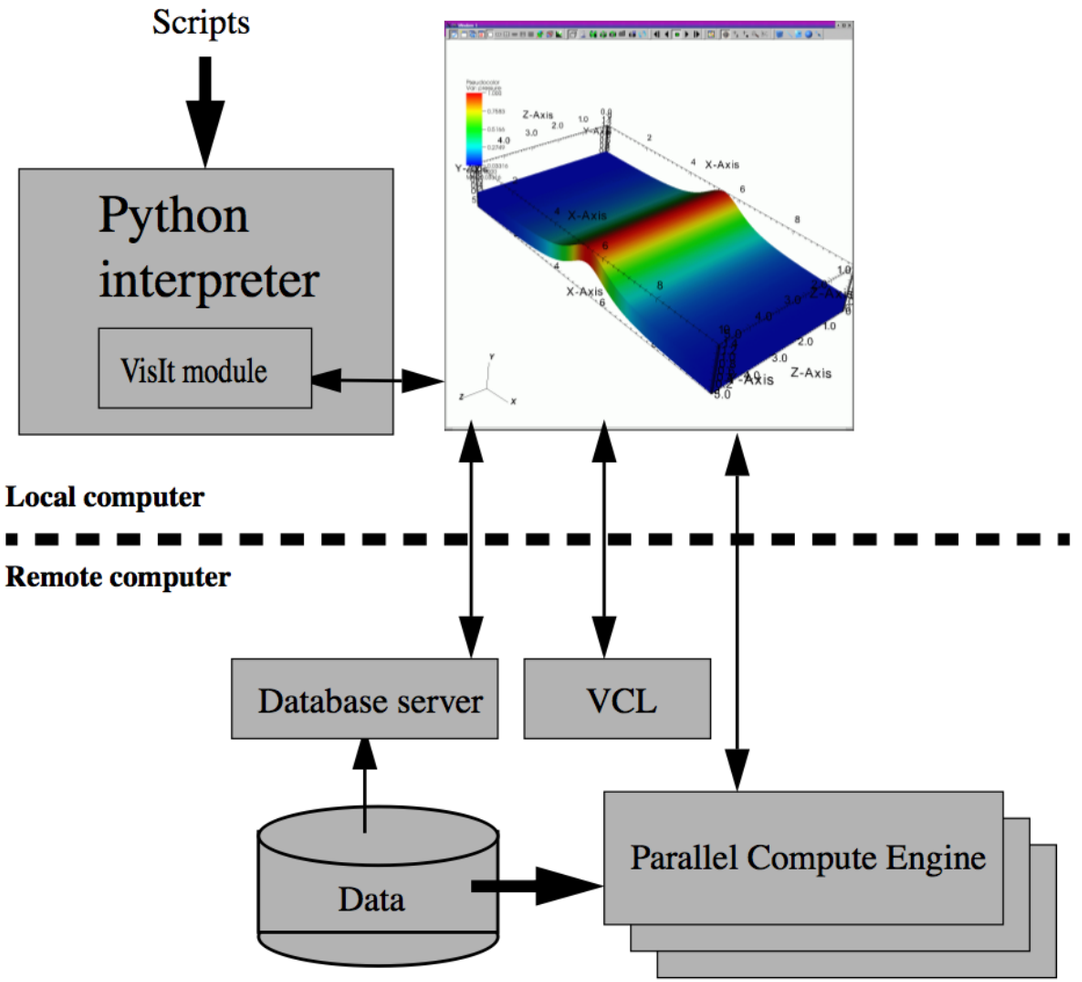


Hands–On Session 2: Part II

Scripting Visualization within VisIt

VisIt and Python

- VisIt can be used from python
  - *import sys*
  - *sys.path.append("/path/to/visit/<version>/<architecture>/lib/site-packages")*
    import visit
  - *visit.Launch()*
- Python can be used within VisIt



Using VisIt GUI and cli Simultaneously

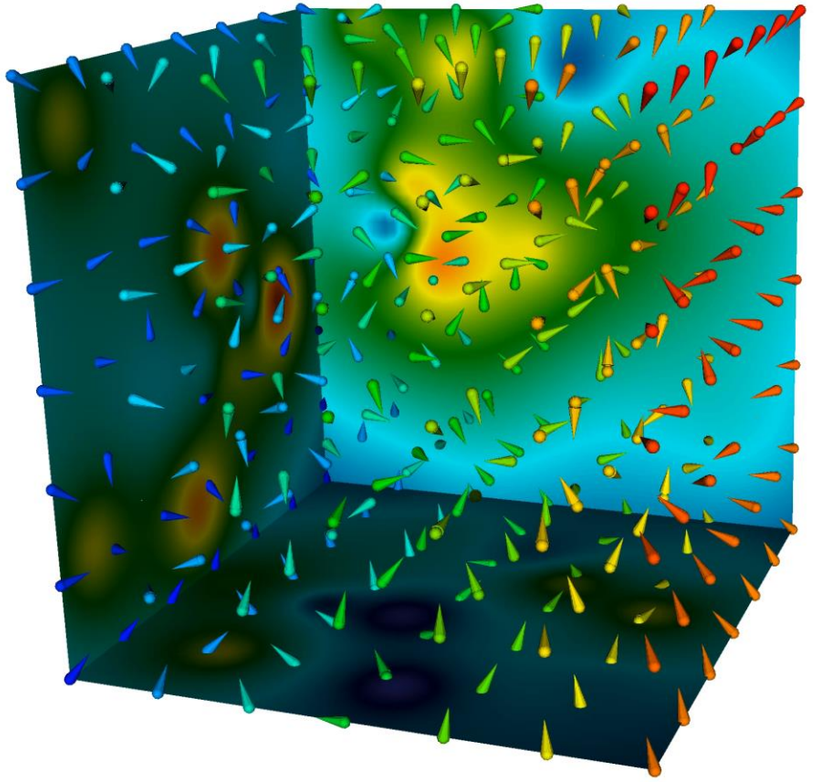
- Open VisIt
  - Open command window: *<Controls/Command>*

- *Go to VisIt Docs*
  - https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/tutorials/Scripting.html
  - We'll walk through some of the initial copy-paste examples

Running a Script Interactively in VisIt

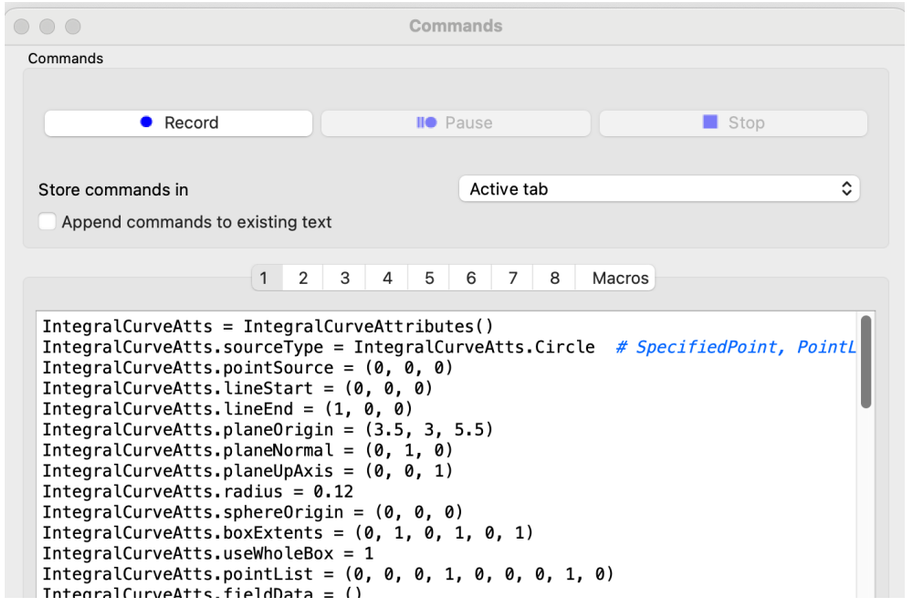
- Works just like the previous examples, but code is in a file
  - Use your favorite editor
  - Have more complicated multi-file scripts

- Enter the following in the cli and click *<Execute>*
  - Source("/path/to/KAUST_Visualization_Vignettes/VisIt_Vignettes/ex04_visitStreamlineAnimation/ex04_visitStreamlineAnimation.py")



Tracing Your Actions

- Open VisIt
  - Open command window: *<Controls/Command>*
  - Open an empty tab
  - Click *<Record>*
  - Interact with the GUI to do the vis you want
  - Click *<Stop>*
  - A lengthy trace will reproduce your actions
    - VisIt prints all possible options for your actions, so you can prune lots of the code away if you are not changing default options



Hands–On Session 3

Scripting Visualization from Command Line

Running Scripts without ParaView GUI

- Navigate to ParaView_Vignettes repo folder on your local computer
  - Run each of the examples (excluding *ex06*, data is too large for local use)
    - cd to individual example directory
    - Use pvpython or pvbatch and run the examples
      - pvbatch and pvpython are the same when running a script, except that pvbatch will run in parallel
      - *pvbatch ex00_pvConeStat.py*
      - *pvpython ex00_pvConeStat.py --verbosity=OFF (hides warnings we don't need)*

```
kressjm@KW-23567:~/Desktop/KAUST_Visualization_Vignettes/ParaView_Vignettes/ex00_pvQuery$ /Applications/ParaView-5.11.1.app/Contents/bin/pvpython ex00_pvConeStat.py
Running ParaView example script:  ex00_pvConeStat.py

Cone Resolution:  6
Cone Height:      1.0
Cone Radius:      0.5
Cone Center:      [0.0, 0.0, 0.0]
Cone Direction:   [1.0, 0.0, 0.0]

Finished ParaView example script

kressjm@KW-23567:~/Desktop/KAUST_Visualization_Vignettes/ParaView_Vignettes/ex00_pvQuery$ /Applications/ParaView-5.11.1.app/Contents/bin/pvbatch ex00_pvConeStat.py
Running ParaView example script:  ex00_pvConeStat.py

Cone Resolution:  6
Cone Height:      1.0
Cone Radius:      0.5
Cone Center:      [0.0, 0.0, 0.0]
Cone Direction:   [1.0, 0.0, 0.0]

Finished ParaView example script
```

Running Scripts without VisIt GUI

- Navigate to VisIt_Vignettes repo folder on your local computer
  - Run each of the examples (excluding *ex06*, data is large for local use)
    - cd to individual example directory
    - *visit -nowin -cli -v 3.3.2 -s ex00 visitQuery.py*

```
kressjm@KW60540:~/packages/KAUST_Visualization_Vignettes/VisIt_Vignettes/ex00_visitQuery$ ../../../visit-install/bin/visit -cli -nowin -s ex00_visitQuery.py
Running: cli3.3.1 -nowin -s ex00_visitQuery.py
Running: viewer3.3.1 -nowin -noint -host 127.0.0.1 -port 5600
Running VisIt example script:  ex00_visitQuery.py

Running script from:  /home/kressjm/packages/KAUST_Visualization_Vignettes/VisIt_Vignettes/ex00_visitQuery
Running script locally, not launching a batch job

Running: mdserver3.3.1 -host 127.0.0.1 -port 5600
Running: /home/kressjm/packages/visit-install/3.3.1/linux-x86_64/bin/mpirun -np 6 /home/kressjm/packages/visit-install/3.3.1/linux-x86_64/bin/engine_par -plugindir /home/kressjm/.visit/3.3.1
/linux-x86_64/plugins:/home/kressjm/packages/visit-install/3.3.1/linux-x86_64/plugins -visithome /home/kressjm/packages/visit-install/3.3.1 -visitarchhome /home/kressjm/packages/visit-instal
l/3.3.1/linux-x86_64 -dir /home/kressjm/packages/visit-install -forcestatic -idle-timeout 480 -noloopback -host KW60540 -port 5600


3D surface area:  The total Surface Area is 2400 parsec^2
Average Value  :  The average value of hardyglobal is 3.27436 Joules
Centroid:         Centroid = (0.205405, 0.162072, -0.0195174)
GridInformation:  Grid 0: type=AVT_RECTILINEAR_MESH, dims={50,50,50}

MinMax:
hardyglobal -- Min = 1.09554 (node 105026 at coord <0.612245, -10, 7.14286>)
hardyglobal -- Max = 5.88965 (node 83943 at coord <7.55102, 1.42857, 3.46939>)


NumNodes:         The actual number of nodes is 125000.
NumZones:         The actual number of zones is 117649.
Volume:           The total Volume is 8000 parsec^3

Finished VisIt example script
```

Demo

Scripting Visualization on Ibex

Access Data on Ibex

- Linux
  - scp –r username@ilogin.ibex.kaust.edu.sa:/path/to/files <local/destination>
  - Mount scratch locally as a folder:
    - sshfs username@mover.ibex.kaust.edu.sa:/ibex/scratch/username <local/destination>
- Mac
  - scp –r username@ilogin.ibex.kaust.edu.sa:/path/to/files <local/destination>
  - Mount scratch locally as a folder:
    - sshfs username@mover.ibex.kaust.edu.sa:/ibex/scratch/username <local/destination>
- Windows
  - scp –r username@ilogin.ibex.kaust.edu.sa:/path/to/files <local/destination>
  - Mount scratch locally as lettered drive:
    - Run SFTP Drive and connect: mover.ibex.kaust.edu.sa; drive path: /ibex/scratch/<username>

Running ParaView Examples on Ibex

- Login to Ibex
  - *ssh -X <username>@glogin.ibex.kaust.edu.sa*
  - *module load paraview*
- Navigate to example dir
  - *cd /ibex/scratch/<username>/KAUST_Visualization_Vignettes/Paraview_Vignettes*
- Run individual examples
  - *cd ex00_pvQuery*
    - *sbatch ex00_ibex_runScript.sbat*
    - View queue info: *squeue –u username*
    - When job finishes view results: *cat ex00.ibex.*.out*
  - *cd ex01_pvScreenshot*
    - *sbatch ex01_ibex_runScript.sbat*
    - When job finishes view image: *display ex01_pvScreenshot.png*
  - *ex02… etc.*

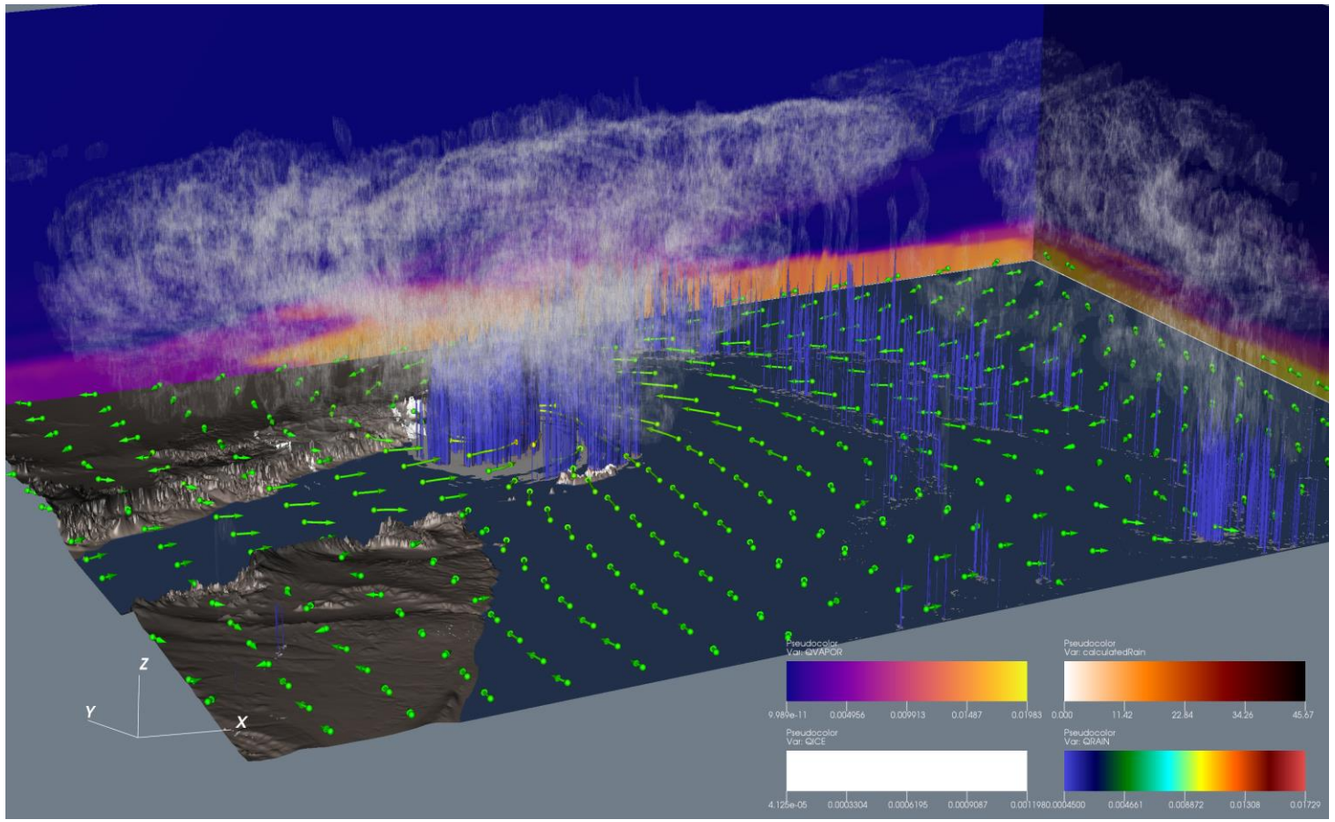
Running VisIt Examples on Ibex

- Login to Ibex
  - *ssh -X <username>@ilogin.ibex.kaust.edu.sa*
  - *module load visit*
- Navigate to example dir
  - *cd /ibex/scratch/<username>/KAUST_Visualization_Vignettes/Visit_Vignettes*
- Run individual examples
  - *cd ex00_visitQuery*
    - *sbatch ex00_ibex_runScript.sbat*
    - View queue info: *squeue –u username*
    - When job finishes view results: *cat ex00.ibex.*.out*
  - *cd ex01_visitScreenshot*
    - *sbatch ex01_ibex_runScript.sbat*
    - When job finishes view image: *display output/ex01_visit000.png*
  - *ex02… etc.*

Parallelizing Bigger Problems

ex06_[pv/visit]LargeData.py

- Run: *`fetchData.sh`* to get data
- Visit
  - 1 node, 1 mpi process
    - real 4m23.610s
  - 8 nodes, 8 mpi processes
    - real 1m59.409s
  - 1 node, 8 mpi processes
    - real 1m46.211s
- ParaView
  - egl
    - 1 node, 1 mpi process, 32 cores 1 gpu
      - real 5m16.889s
    - 1 node, 1 mpi process, 32 cores 2 gpu
      - real 4m56.753s
  - mesa: don't volume render with mesa, transfer functions don't work properly, and it is slow
    - ~18 minutes on a beefy machine



~5 GB data files, expands to ~15 GB when loaded in VisIt/ParaView. The pipeline for this image uses ~59 GB of RAM

! **Note:** VisIt/ParaView timings are not directly comparable. VisIt is not using volume rendering, nor does it make use of GPU. Timings are meant to show how resources affect time-to-solution

Wrap-up

Best Practices

How do I use ParaView or VisIt?

- If your data is small/manageable
  - Do your visualizations on your laptop, desktop, or IT Remote Workstation
- If your data is medium/large
  - Do interactive visualization on Ibex
    - Run it on your local machine and connect directly to Ibex to load/process/visualize
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/ParaView_Vignettes#using-paraview-interactively-on-ibex
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/VisIt_Vignettes#using-visit-interactively-on-ibex
- If your data is large/huge and you have a defined workflow
  - Do batch visualization on Shaheen
  - Look at example scripts and talk to us if you have questions
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/VisIt_Vignettes#expy
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/ParaView_Vignettes#expy
- If you have repeatable repetitive tasks
  - Do scripted or batch visualization

Thanks!

Contacts:

james.kress@kaust.edu.sa

help@vis.kaust.edu