Leveraging Norwegian AI Cloud (NAIC) for High Performance Visualization

An Introduction to using ParaView

Sherin Sugathan | dScience | UiO

Norwegian AI Cloud (NAIC)

Establishing powerful AI/ML infrastructure in Norway

The NAIC **provides access to High-Performance Computing (HPC) resources** and **support to researchers, PhD students, educators, entrepreneurs, startups, and SMEs** interested in exploring AI methodologies, as well as developing advanced AI models and algorithms.[1]

UNIVERSITETET I OSLO

NORA.ai

NTNU | Norwegian University of Science and Technology

UNIVERSITETET I BERGEN

simula

UiA University of Agder Norway

SINTEF

UiT The Arctic University of Norway

sigma2

NORCE

Norwegian AI Cloud

[1] Please visit https://www.naic.no/ for more details

sherin.sugathan@dscience.uio.no

Outline

Introduction to ParaView

Python Scripting in ParaView

Extending ParaView

ParaView under NAIC

Advanced Topics

High Performance Computing

User Requirements

"The data don’t fit in my memory. Even if it did, the computations would still take months if I used my local resources."

NAIC User Survey 2023

- ”I think it is hard to speed up the computations, and I also don’t care whether running my code takes 14 hours or 15 hours. What is crucial for me is to easily check the results once the computations are finished. To do so, I often need to display images, open files etc. This slows me down considerably, as I always need to set up a workaround to solve the issue” - User needs solution to display/render data without copying it back to the client.

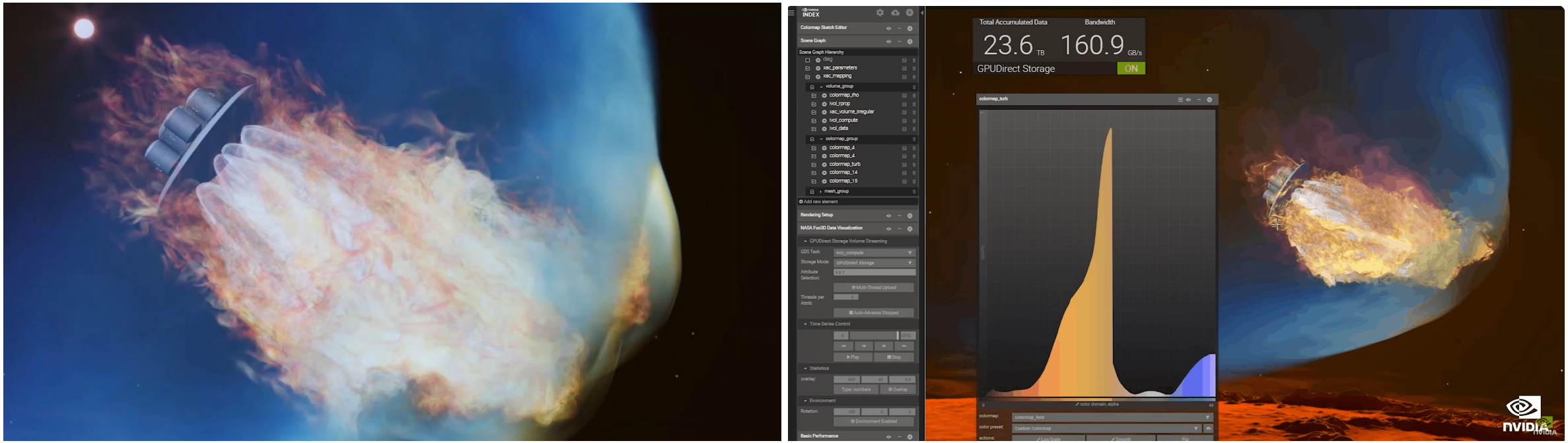
What will the user do if is image is tooooooooooooo big to open?

High Performance Visualization

Relevance of HPC in Visualization Tasks

How NASA Is Helping Humans Reach the Red Planet, Using GPUs and 3D Visualization

Each Simulation producing 200 TB (and thousands of these representing possible conditions)
Using Summit HPC, simulation runtime reduced from months to a single week.
Interactive Visualizations were developed to study the simulation data.



Original article:
https://developer.nvidia.com/blog/isc20-featured-demo-visualizing-150-terabytes-of-data/
Video Source:
https://www.youtube.com/watch?v=GAZP1NcdWMo

sherin.sugathan@dscience.uio.no

ParaView

About the tool

python	VisTrails	MeshLab	Ensight
Gmsh	OpenDX	ParaView	Glumpy
Blender	Cinema	VisIt	MayaVI

ParaView

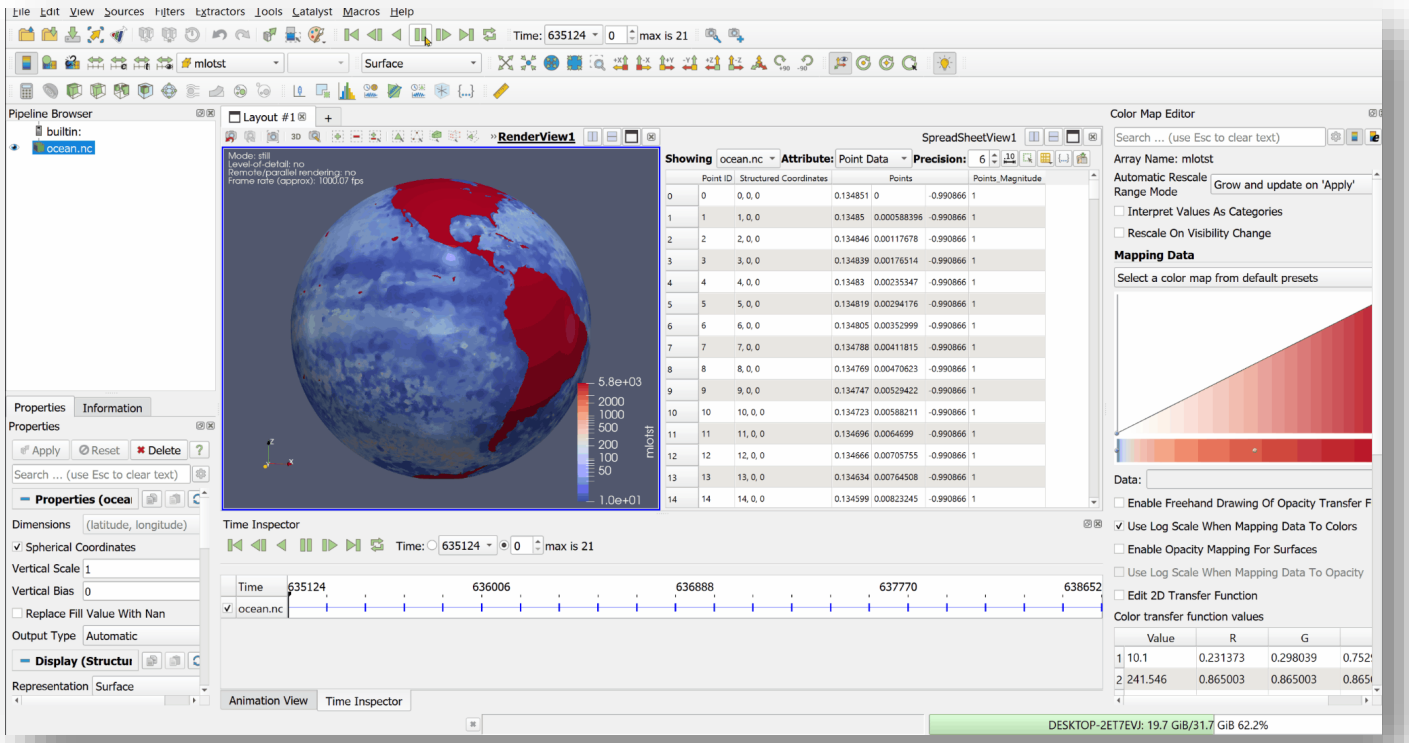
About the tool

python	VisTrails	MeshLab	Ensight
Gmsh	OpenDX	ParaView	Glumpy
Blender	Cinema	VisIt	MayaVI

sherin.sugathan@dscience.uio.no

ParaView

About the tool



- Open-source and multiplatform
- Frontend to VTK+Qt
- User Interface
- Cross Platform Compatibility
- Large data handling
- Large userbase
- Extensible Architecture
- Rich Visualization Techniques
- Scriptable
- Community and Support

An open-source visualization **application** and **architecture** for visual analysis of massive scientific datasets.

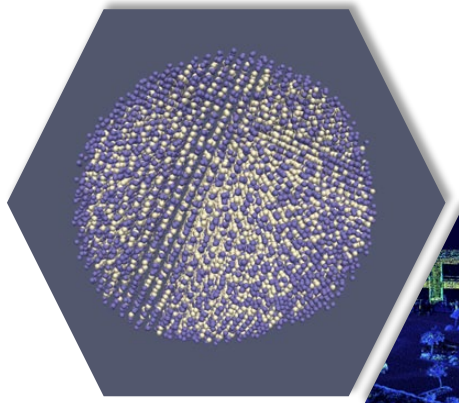
Reference:
http://www.paraview.org

sherin.sugathan@dscience.uio.no

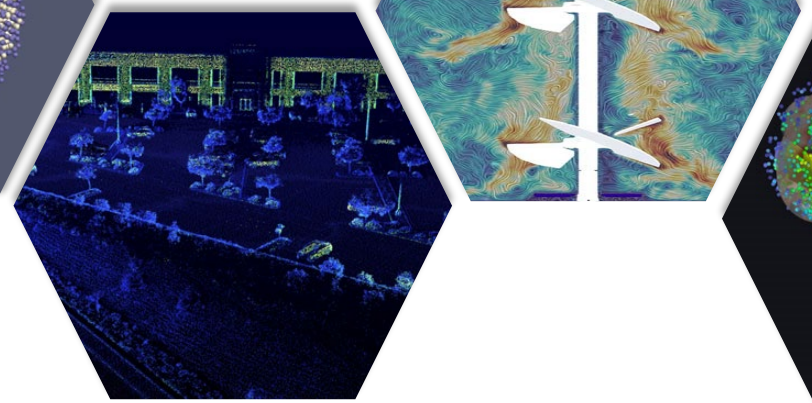
ParaView

Application Areas

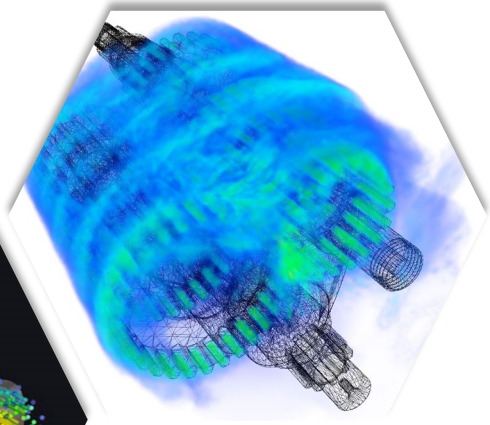
Material Science



Computational Fluid Dynamics



Engineering



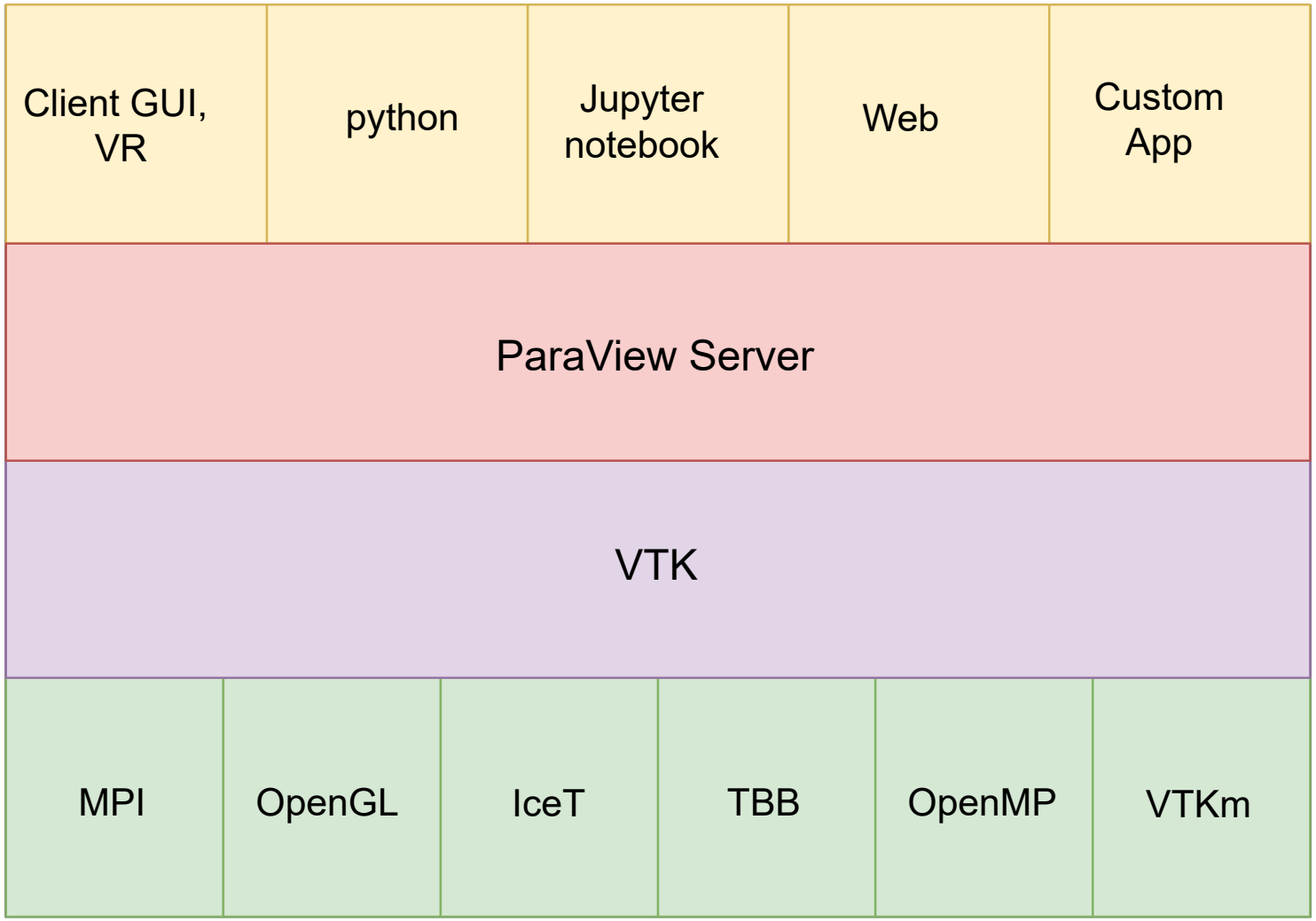
etc. . . .

Sensor Data

Medical Science

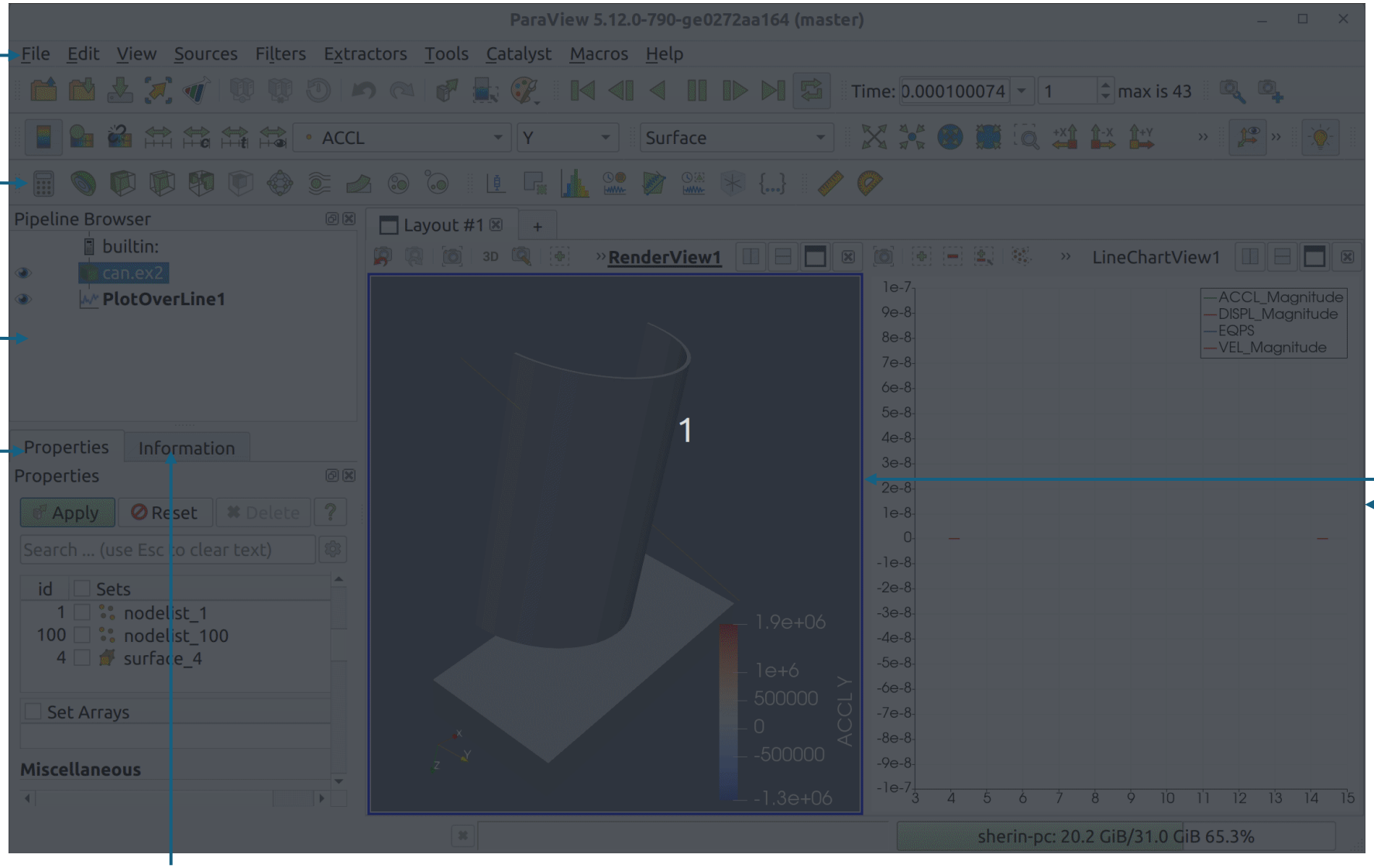
ParaView

Application Architecture



Basic Usage of ParaView

The User Interface



sherin.sugathan@dscience.uio.no

Basic Usage of ParaView

Data Formats

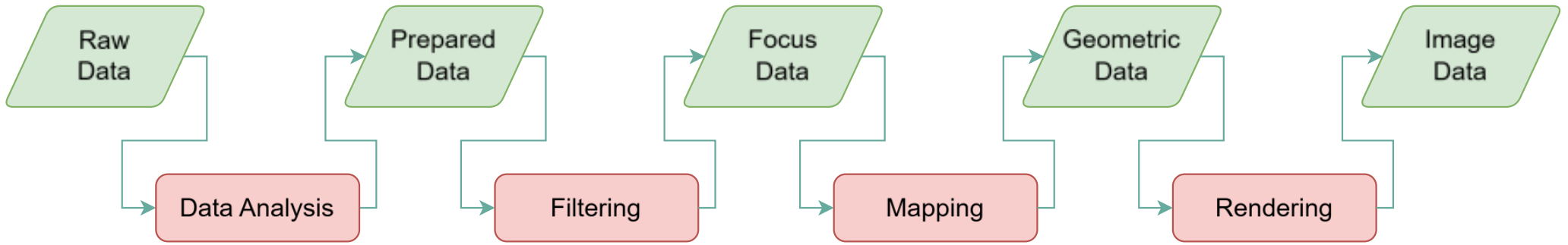
ParaView Data (.pvd)
VTK (.vtp, .vtu, .vti, .vts, .vtr)
VTK Legacy (.vtk)
VTK Multi Block (.vtm,.vtmb,.vtmg,.vthd,.vthb)
Partitioned VTK (.pvtu, .pvti, .pvts, .pvtr)
ADAPT (.nc, .cdf, .elev, .ncd)
ANALYZE (.img, .hdr)
ANSYS (.inp)
AVS UCD (.inp)
BOV (.bov)
BYU (.g)
CAM NetCDF (.nc, .ncdf)
CCSM MTSD (.nc, .cdf, .elev, .ncd)
CCSM STSD (.nc, .cdf, .elev, .ncd)
CEAucd (.ucd, .inp)
CGNS (.cgns)
CMAT (.cmat)
CML (.cml)
CTRL (.ctrl)
Chombo (.hdf5, .h5)
Claw (.claw)
Comma Separated Values (.csv)
Cosmology Files (.cosmo, .gadget2)
Curve2D (.curve, .ultra, .ult, .u)
DDCMD (.ddcmd)
Digital Elevation Map (.dem)
Dyna3D(.dyn)
EnSight (.case, .sos)
Enzo boundary and hierarchy
ExodusII (.g, .e, .exe, .ex2, .ex2v.., etc)
ExtrudedVol (.exvol)
FVCOM (MTMD, MTSD, Particle, STSD)
Facet Polygonal Data
Flash multiblock files
Fluent Case Files (.cas)
GGCM (.3df, .mer)
GTC (.h5)
GULP (.trg)
Gadget (.gadget)
Gaussian Cube File (.cube)
JPEG Image (.jpg, .jpeg)
LAMPPS Dump (.dump)
LAMPPS Structure Files
LODI (.nc, .cdf, .elev, .ncd)
LODI Particle (.nc, .cdf, .elev, .ncd)
LS-DYNA (.k, .lsdyna, .d3plot, d3plot)
M3DCl (.h5)
MFIX Unstructred Grid (.RES)
MM5 (.mm5)
MPAS NetCDF (.nc, .ncdf)
Meta Image (.mhd, .mha)
Miranda (.mir, .raw)
Multilevel 3d Plasma (.m3d, .h5)
NASTRAN (.nas, .f06)
Nek5000 Files
Nrrd Raw Image (.nrrd, .nhdr)
OpenFOAM Files (.foam)
PATRAN (.neu)
PFLOTRAN (.h5)
PLOT2D (.p2d)
PLOT3D (.xyz, .q, .x, .vp3d)
PLY Polygonal File Format
PNG Image Files
POP Ocean Files
ParaDIS Files
Phasta Files (.pht)
Pixie Files (.h5)
ProSTAR (.cel, .vrt)
Protein Data Bank (.pdb, .ent, .pdb)
Raw Image Files
Raw NRRD image files (.nrrd)
SAMRAI (.samrai)
SAR (.SAR, .sar)
SAS (.sasgeom, .sas, .sasdata)
SESAME Tables
SLAC netCDF mesh and mode data
SLAC netCDF particle data
Silo (.silo, .pdb)
Spheral (.spheral, .sv)
SpyPlot CTH
SpyPlot (.case)
SpyPlot History (.hscth)
Stereo Lithography (.stl)
TFT Files
TIFF Image Files
TSurf Files
Tecplot ASCII (.tec, .tp)
Tecplot Binary (.plt)
Tetrad (.hdf5, .h5)
UNIC (.h5)
VASP CHGCA (.CHG)
VASP OUT (.OUT)
VASP POSTCAR (.POS)
VPIC (.vpc)
VRML (.wrl)
Velodyne (.vld, .rst)
VizSchema (.h5, .vsh5)
Wavefront Polygonal Data (.obj)
WindBlade (.wind)
XDMF and hdf5 (.xmf, .xdmf)
XMol Molecule

+ any custom datasets loaded through custom sources/plugins

sherin.sugathan@dscience.uio.no

Basic Usage of ParaView

The Visualization Pipeline



sherin.sugathan@dscience.uio.no

ParaView

Building Visual Representations

Filters **transform input data and produce results** on their outputs.

- Each filter has a **fixed number of input and output ports**.

•Input ports accept data for specific roles within the filter.

- •Example: "Resample With Dataset" filter has two input ports:
  - •"Input" for the dataset providing attributes to interpolate.
  - •"Source" for the dataset used as the mesh for re-sampling.

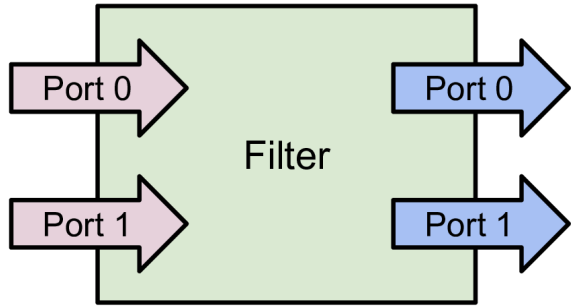
•Each **input port can optionally accept multiple input connections**.

- •Example: "Append Datasets" filter has one input port (named "Input") that can accept multiple dataset connections.

•Filters define whether an input port can accept one or many input connections.

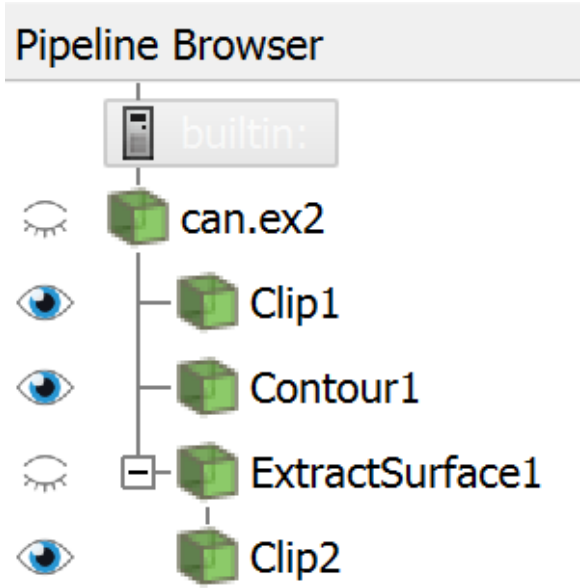
•Filter **properties allow control over the filtering algorithm**.

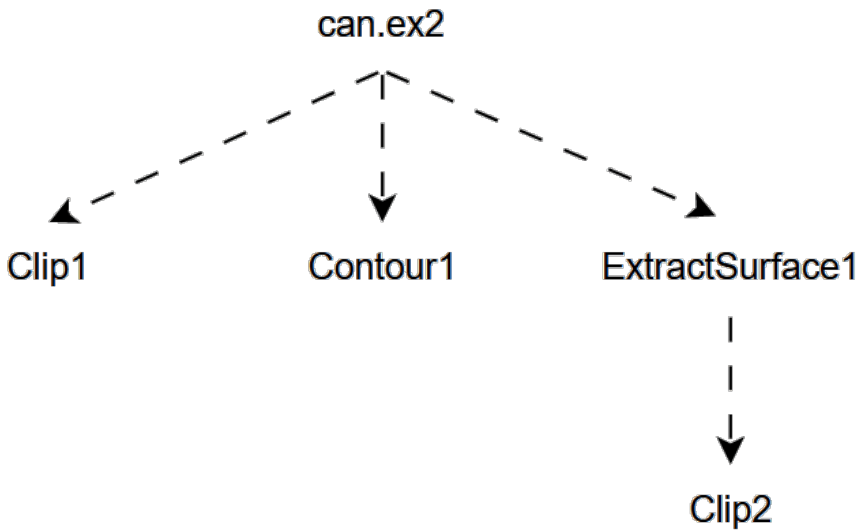
•Available properties depend on the specific filter.



ParaView

Building Visual Representations

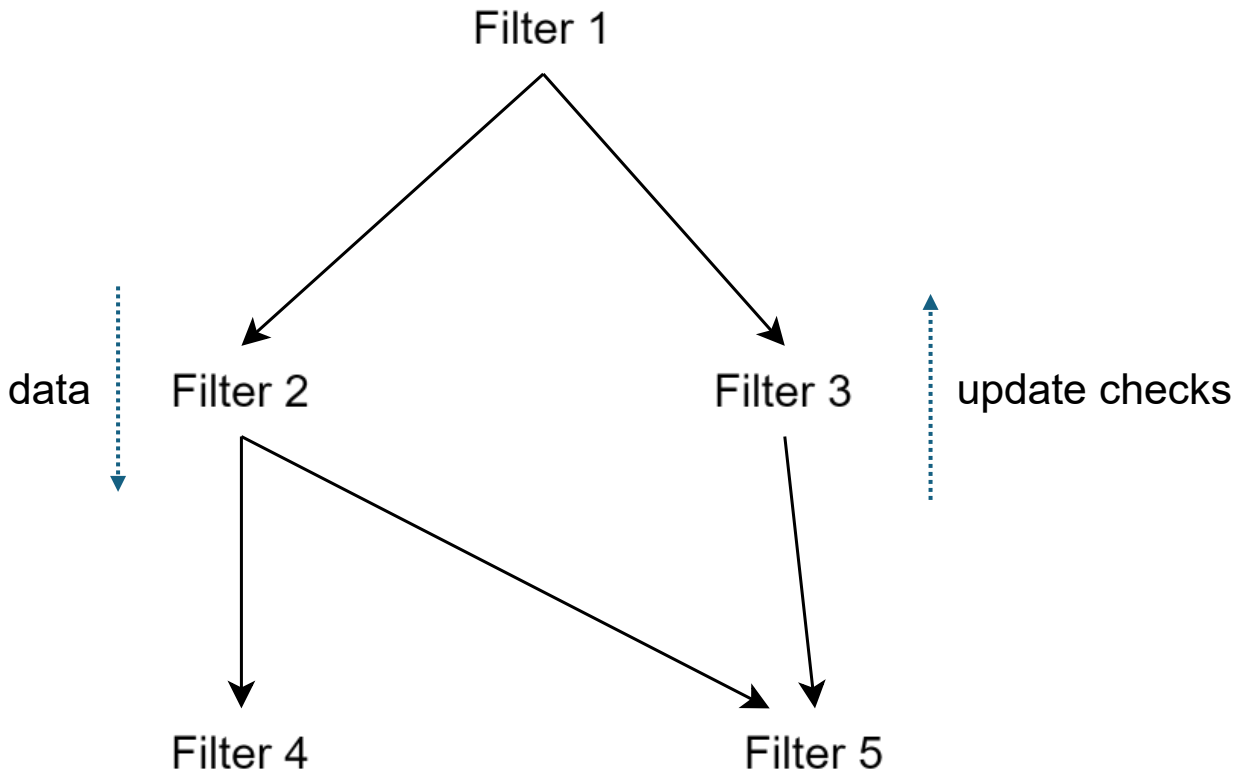




sherin.sugathan@dscience.uio.no

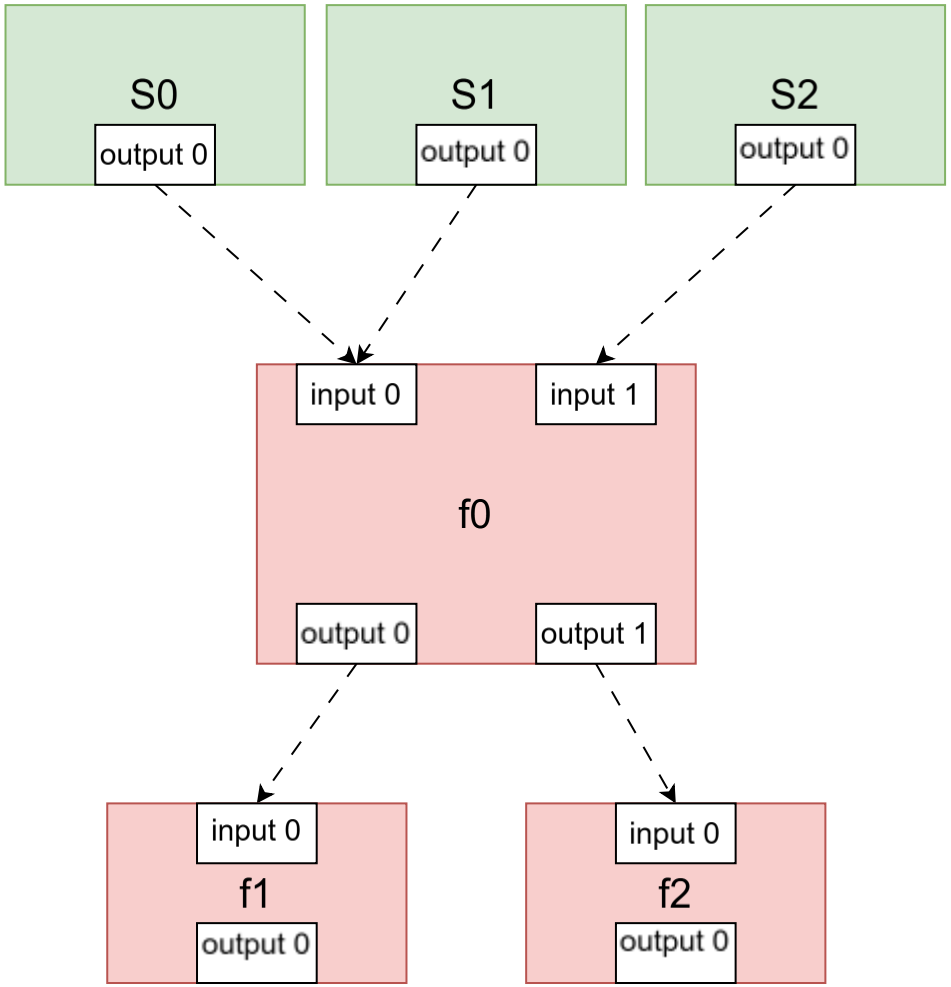
ParaView

Building Visual Representations



ParaView

Building Visual Representations




```
s0 = Source()
s1 = Source()
s2 = Source()

f0 = Filter0()
f0.AddInputConnection(0, s0.GetOutputPort())
f0.AddInputConnection(0, s1.GetOutputPort())
f0.SetInputConnection(1, s2.GetOutputPort())

f1 = Filter1()
f1.SetInputConnection(f0.GetOutputPort(0))

f2 = Filter1()
f2.SetInputConnection(f0.GetOutputPort(1))
```

Reference:
https://www.kitware.com/a-vtk-pipeline-primer-part-1/

sherin.sugathan@dscience.uio.no

Basic Usage of ParaView

A quick walkthrough

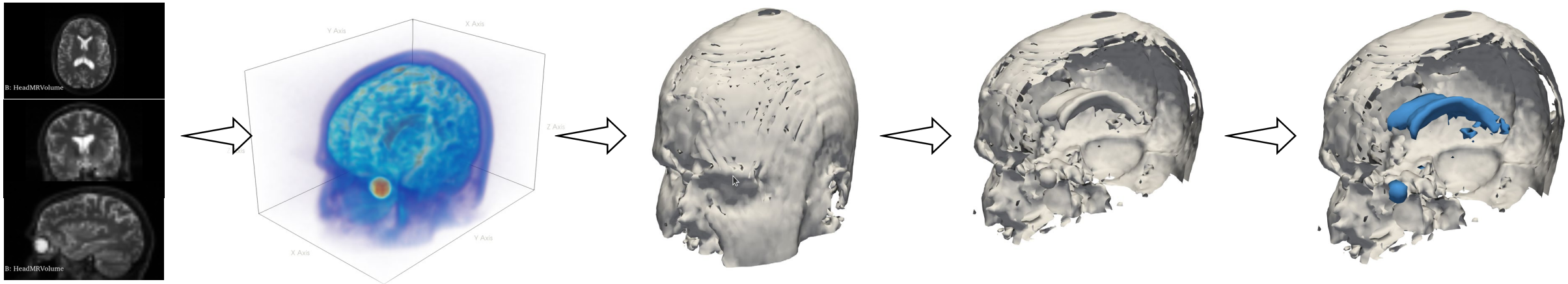
demo

sherin.sugathan@dscience.uio.no

Basic Usage of ParaView

Example: Medical Datasets

MetaImage → MetaFileSeriesReader → 3D grid → Contour → Polygon data → Clip → Polygon data → Mapping + Render → Image



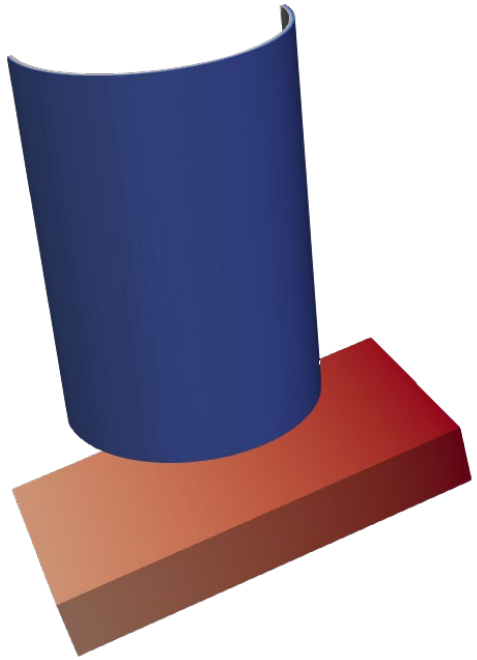
Data file location:
workshop_2024/exercise/basics/HeadMRVolume.mhd

sherin.sugathan@dscience.uio.no

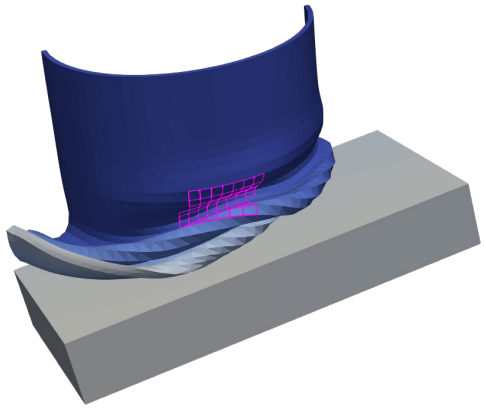
Basic Usage of ParaView

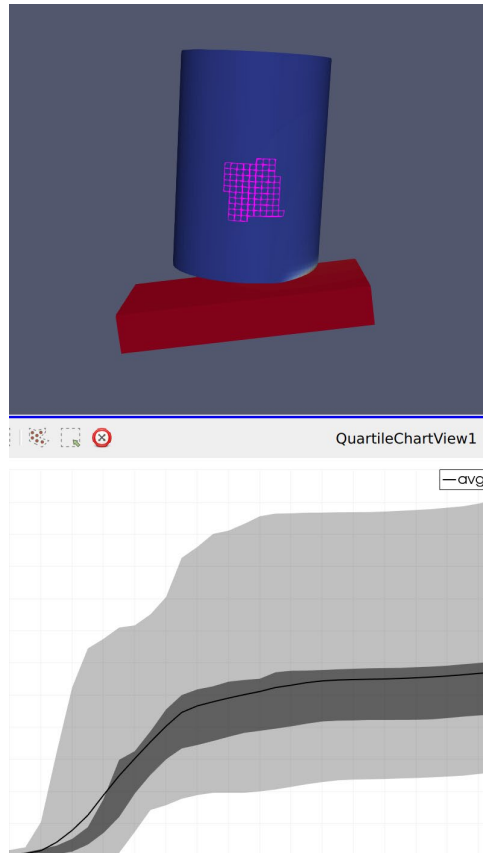
Example: Finite Element Analysis

ExodusIIReader



Plot Selection Over Time





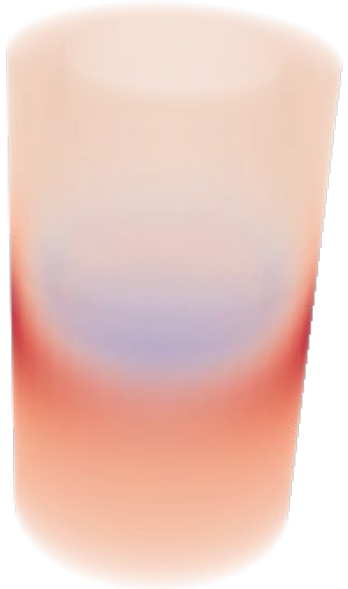
Data file location:
workshop_2024/exercise/basics/can.ex2

sherin.sugathan@dscience.uio.no

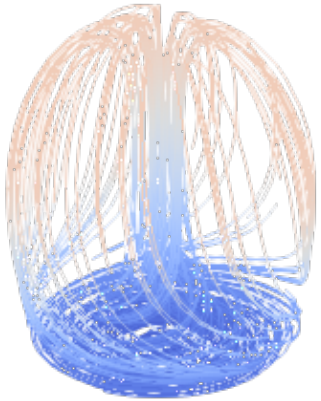
Basic Usage of Preview

Example: StreamTracing

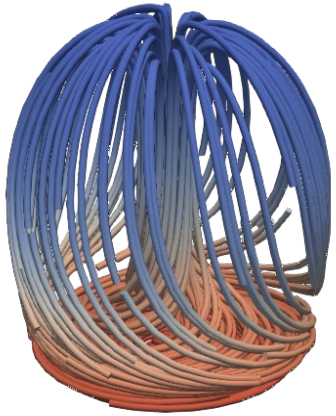
ExodusIIReader



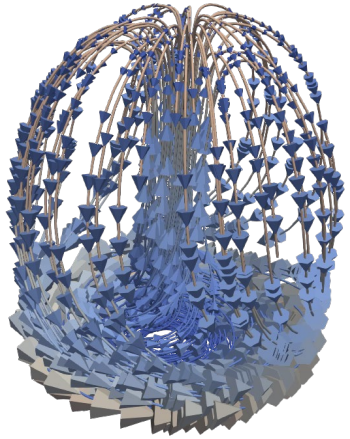
StreamTracer



Tube



Glyph

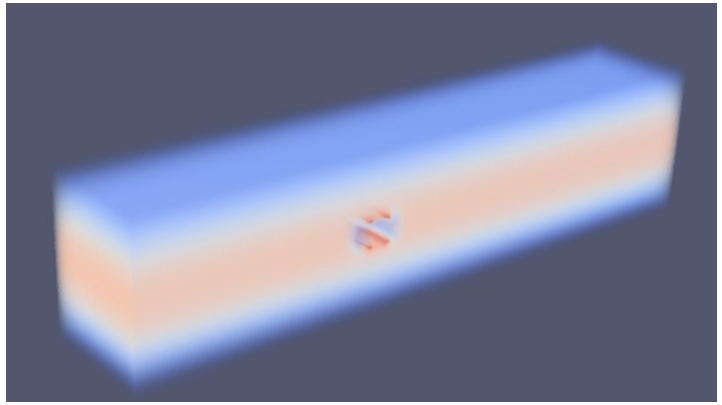


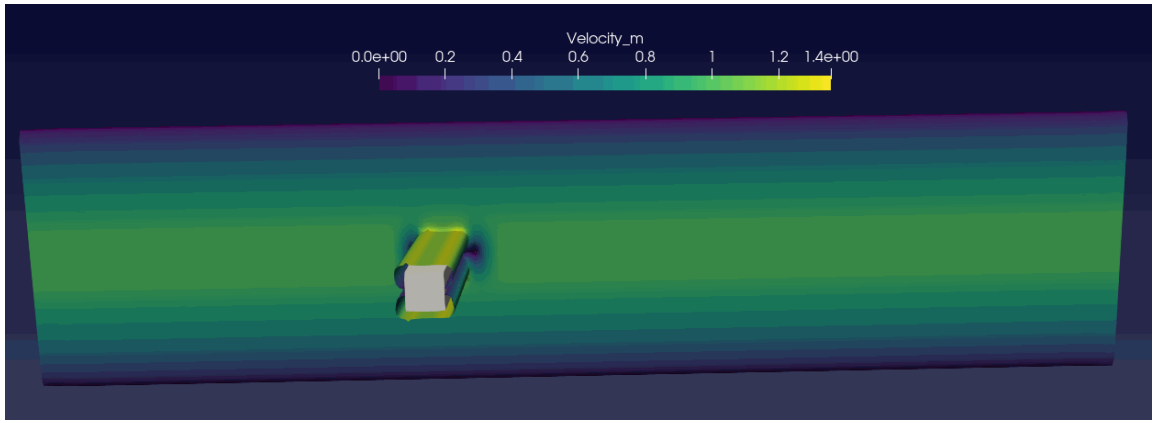
Data file location:
workshop_2024/exercise/basics/disk_out_ref.ex2

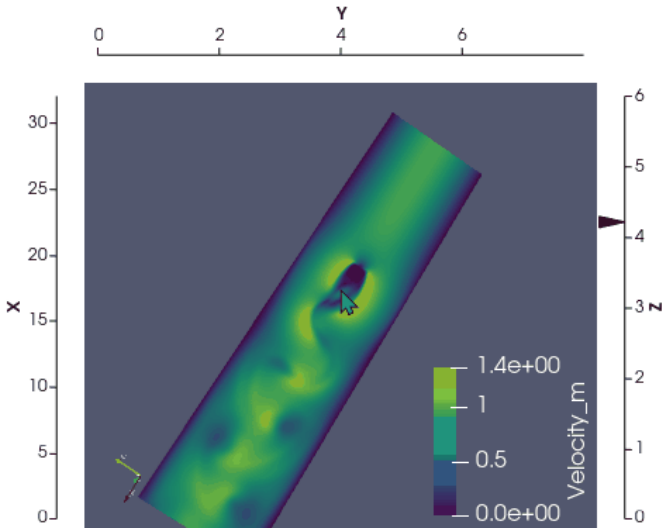
sherin.sugathan@dscience.uio.no

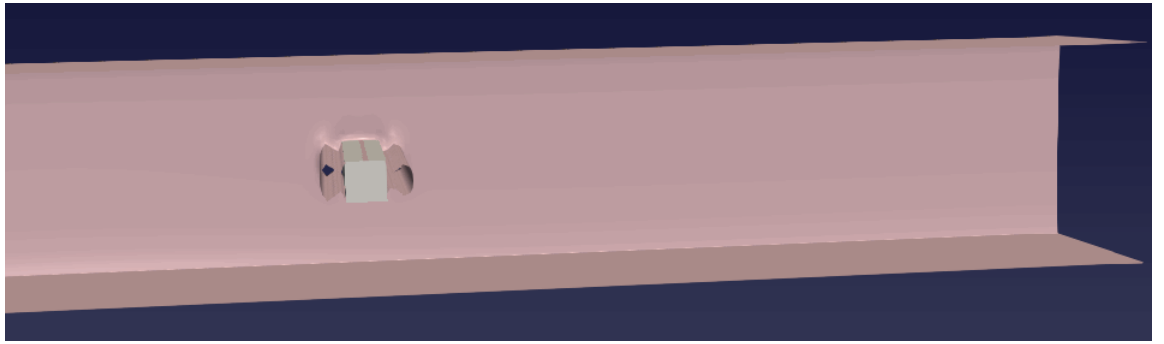
Basic Usage of ParaView

Example: Flow Visualization







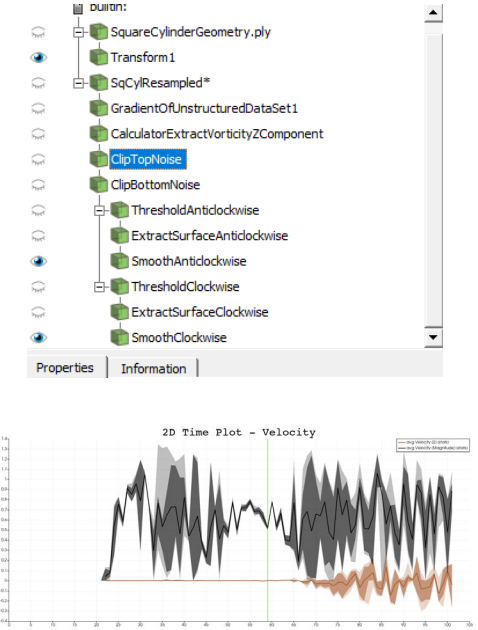


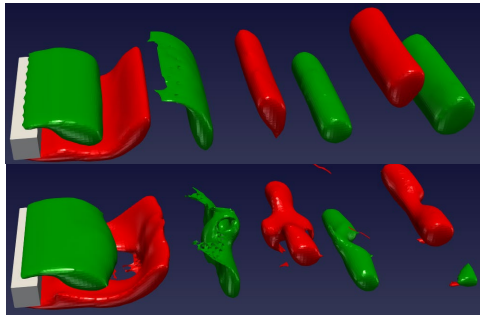
Flow dataset can be downloaded from:
https://cgl.ethz.ch/research/visualization/data.php

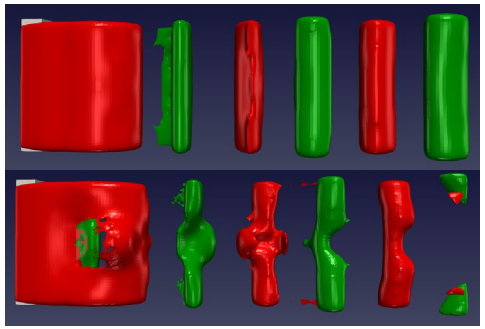
sherin.sugathan@dscience.uio.no

Basic Usage of ParaView

Example: Flow Visualization







Flow dataset can be downloaded from:
https://cgl.ethz.ch/research/visualization/data.php

sherin.sugathan@dscience.uio.no

Basic Usage of ParaView

Compatibility of Filters

AMR Connectivity
AMR Contour
AMR Dual Clip
AMR Fragment Integration
AMR Fragments Filter
Adaptive Resample To Image
Add Field Arrays
Aggregate Dataset
Align Image Origins
Angular Periodic Filter
Animate Modes
Annotate Attribute Data
Annotate Global Data
Annotate Selection
Annotate Time Filter
Append Arc-Length
Append Attributes
Append Datasets
Append Geometry
Append Location Attributes
Append Molecule
Append Reduce
Block Scalars
Boundary Mesh Quality
Bounding Ruler
Calculator
Cell Centers
Cell Data to Point Data
Cell Size
Clean
Clean Cells to Grid
Clean to Grid
Clip
Clip Closed Surface
Clip Generic Dataset
Compute Connected Surface Properties
Compute Derivatives
Compute Molecule Bonds
Compute Quartiles
Compute Surface Sides
Connectivity
Contingency Statistics
Contour
Contour Generic Dataset

Convert AMR dataset to Multi-block
Convert Into Molecule
Convert Polyhedral Cells
Convert To Cell Grid
Convert To MultiBlock
Convert To PartitionedDataSetCollection
Convert To Point Cloud
Count Cell Faces
Count Cell Vertices
Curvature
D3
Date To Numeric
Decimate
Decimate Polyline
Deflect Normals
Delaunay 2D
Delaunay 3D
Descriptive Statistics
Distribute Points
Elevation
Environment Annotation
Equalizer Filter
Evenly Spaced Streamlines 2D
Extract AMR Blocks
Extract Block
Extract CTH Parts
Extract Cells Along Line
Extract Cells Along Lines Custom Source
Extract Cells By Region
Extract Cells By Type
Extract Component
Extract Edges
Extract Enclosed Points
Extract Generic Dataset Surface
Extract Ghost Cells
Extract Location
Extract Particles Over Time
Extract Region Surface
Extract Selection
Extract Subset
Extract Subset With Seed
Extract Surface
Extract Time Steps
FFT Of Selection Over Time

Feature Edges
Field Data to Attribute
Finite Element Field Distributor
Force Time
Gaussian Resampling
Generate Global Ids
Generate Ids
Generate Process Ids
Generate Quadrature Points
Generate Quadrature Scheme Dictionary
Generate Spatio Temporal Harmonics
Generate Surface Normals
Generate Surface Tangents
Generate Time Steps
Ghost Cells Generator
Glyph
Glyph With Custom Source
Gradient
Group Datasets
Group Time Steps
Histogram
Histogram 2D
HyperTreeGrid Axis Reflection
HyperTreeGrid Cell Centers
HyperTreeGrid Depth Limiter
HyperTreeGrid Evaluate Coarse
HyperTreeGrid Geometry Filter
HyperTreeGrid Ghost Cells Generator
HyperTreeGrid To Dual Grid
HyperTreeGrid To UnstructuredGrid
Image Data To AMR
Image Data To Uniform Grid
Image Data to Point Set
Integrate Variables
Interpolate to Quadrature Points
Intersect Fragments
Iso Volume
K Means
Level Scalars(Overlapping AMR)
Linear Cell Extrusion
Linear Extrusion
Loop Subdivision
Mask Points
Material Interface Filter

Median
Merge Blocks
Merge Time
Merge Vector Components
Mesh Quality
Molecule To Lines
Multicorrelative Statistics
Normal Glyphs
OME TIFF Channel Calculator
Octree Image to PointSet
Outline
Outline Corners
Outline Curvilinear DataSet
Overlapping Cells Detector
PCA Normal Estimation
Particle Path
Particle Tracer
Partition Balancer
Pass Arrays
Perlin Noise
Plot Data
Plot Data Over Time
Plot Global Variables Over Time
Plot On Intersection Curves
Plot On Sorted Lines
Plot Over Line
Plot Over Lines From Custom Source
Plot Selection Over Time
Point Data to Cell Data
Point Dataset Interpolator
Point Line Interpolator
Point Plane Interpolator
Point Volume Interpolator
PointSet To Octree Image
Principal Component Analysis
Probe Location
Process Id Scalars
Programmable Annotation
Programmable Filter
Python Annotation
Python Calculator
Quadric Clustering
Random Attributes
Random Vectors

Rectilinear Data to Point Set
Rectilinear Grid Connectivity
Redistribute DataSet
Reflect
Remove Ghost Information
RenameArrays
Resample AMR
Resample To Image
Resample To Line
Resample With Dataset
Reverse Sense
Ribbon
Rotational Extrusion
SPH Dataset Interpolator
SPH Line Interpolator
SPH Plane Interpolator
SPH Volume Interpolator
Scatter Plot
Shrink
Slice
Slice Along PolyLine
Slice Generic Dataset
Slice With Plane
Smooth
Streak Line
Stream Tracer
Stream Tracer For Generic Datasets
Stream Tracer With Custom Source
Subdivide
Surface Flow
Surface Vectors
Synchronize Time
Table FFT
Table To Points
Table To Structured Grid
Temporal Array Operator
Temporal Cache
Temporal Interpolator
Temporal Particles To Pathlines
Temporal Shift Scale
Temporal Snap-to-Time-Step
Temporal Statistics
Tensor Glyph
Tensor Principal Invariants

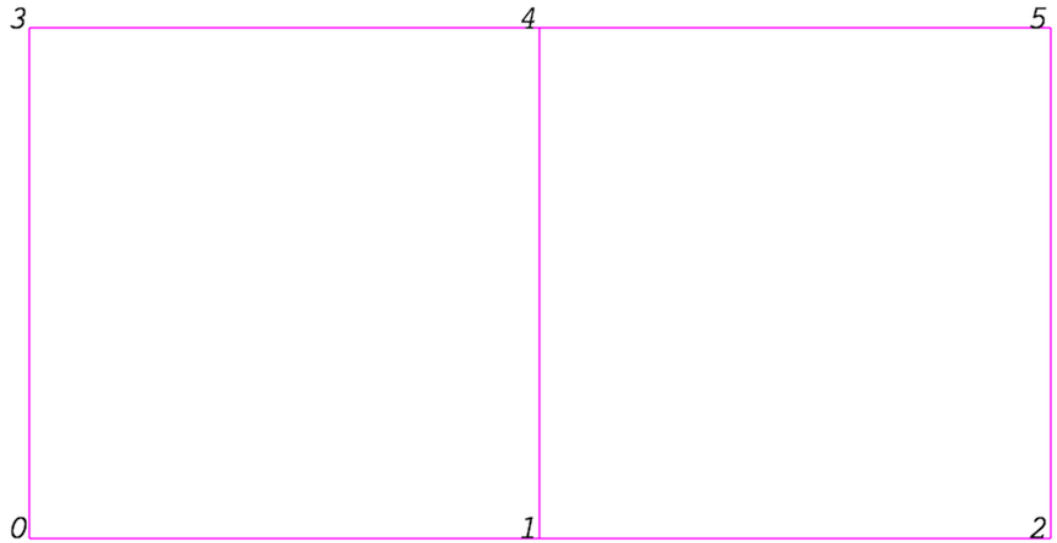
Tessellate
Tetrahedralize
Texture Map to Cylinder
Texture Map to Plane
Texture Map to Sphere
Threshold
Threshold Table
Time Step Progress Bar
Transform
Transpose Table
Triangle Strips
Triangulate
Tube
Validate Cells
Volume Of Revolution
Vortex Cores
Warp By Scalar
Warp By Vector
Yield Criteria
Youngs Material Interface

sherin.sugathan@dscience.uio.no

Understanding Data

The VTK data model

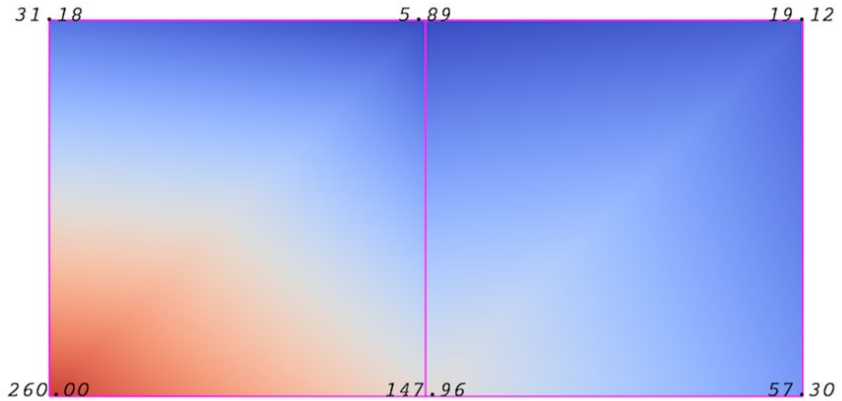
Points are the coordinates in space that represent vertices.
Cells are the topological elements that define shapes using these points.
Attributes can be associated with both points and cells for additional data representation.

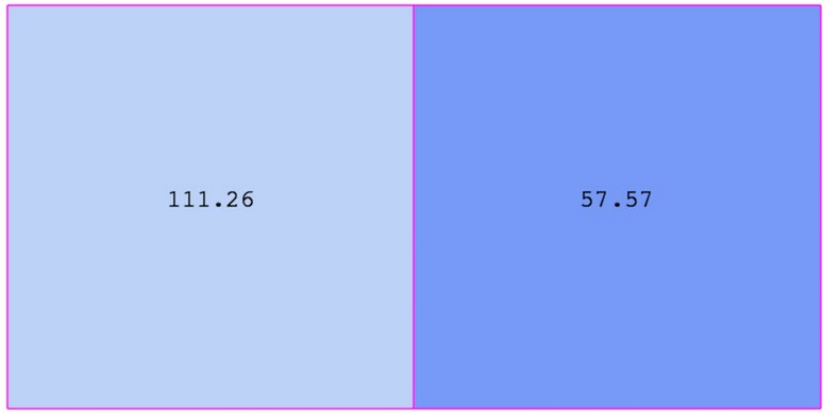


Understanding Data

The VTK data model

Points are the coordinates in space that represent vertices.
Cells are the topological elements that define shapes using these points.
Attributes can be associated with both points and cells for additional data representation.

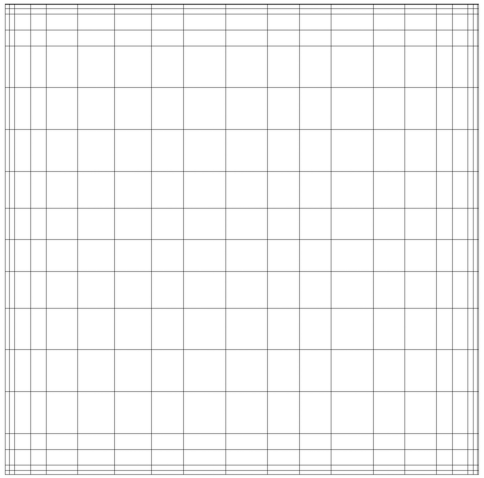




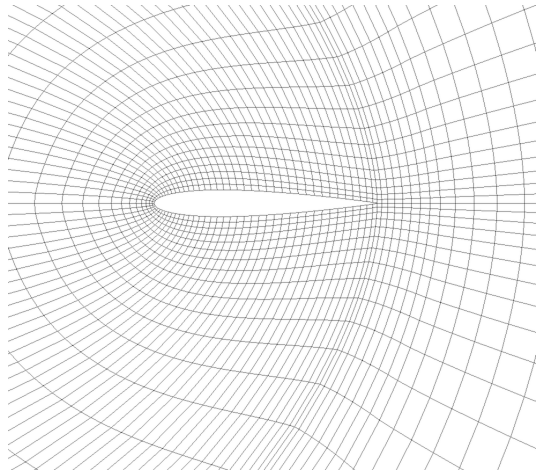
Understanding Data

The VTK data model

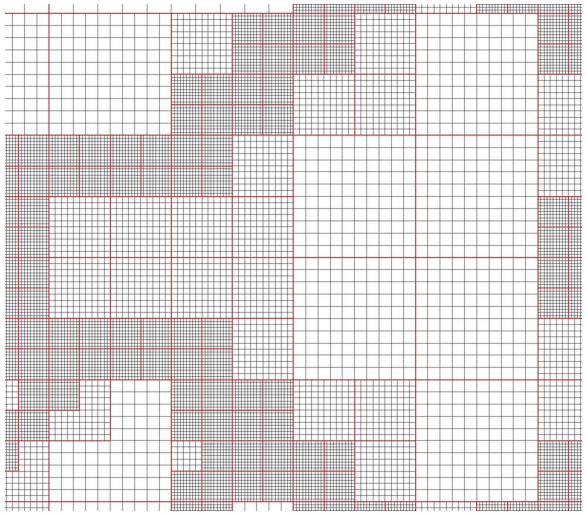
Uniform rectilinear grid



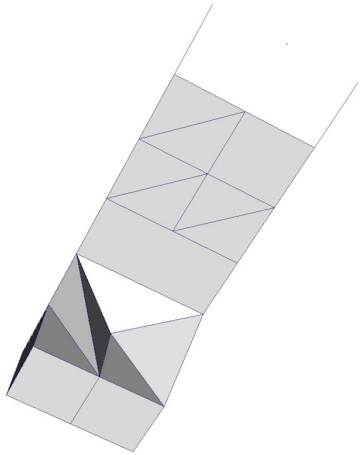
Rectilinear Grid



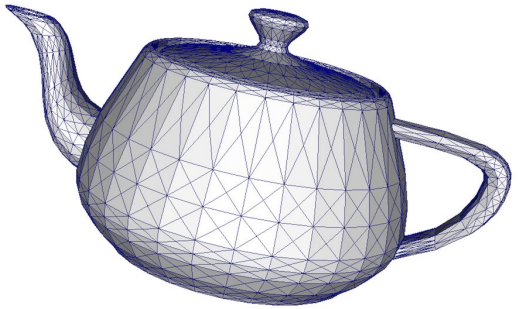
Curvilinear grid



AMR dataset



Unstructured grid



Polygonal grid

sherin.sugathan@dscience.uio.no

Python Scripting in ParaView

Why use Python in ParaView?

Automate repetitive tasks.

Customize and extend functionality.

Integrate ParaView with other python libraries.

sherin.sugathan@dscience.uio.no

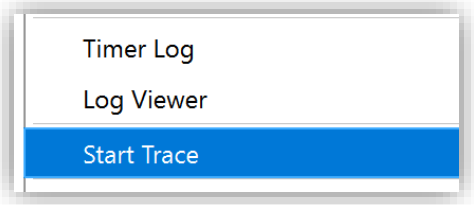
Python Scripting in ParaView

Leveraging Python for Automation

Built-in python shell

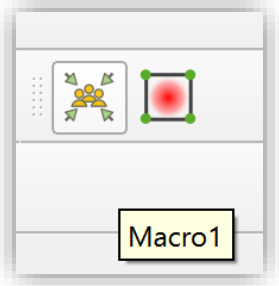
View → Python Shell

Trace Recorder



Tools → Start Trace
Tools → Stop Trace

Macros



Tools → Python Script Editor

demo

ParaView's Python documentation at:
https://kitware.github.io/paraview-docs/latest/python/

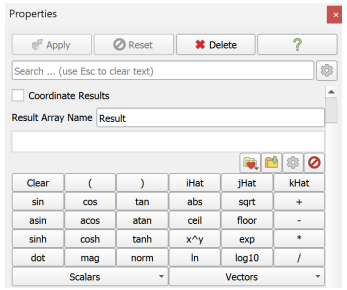
sherin.sugathan@dscience.uio.no

Python Scripting in ParaView

Leveraging Python for Automation

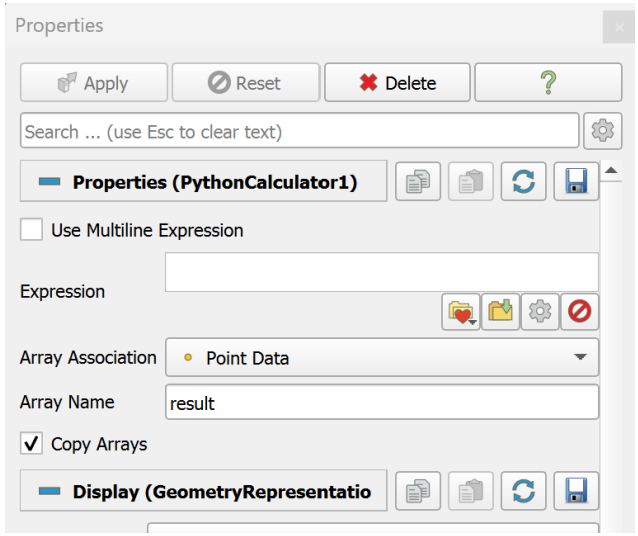
Calculator

/, *, -, +, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, min, max, x^y, sqrt, exp, ln, log10, ceil, floor, abs, dot, cross, mag, norm



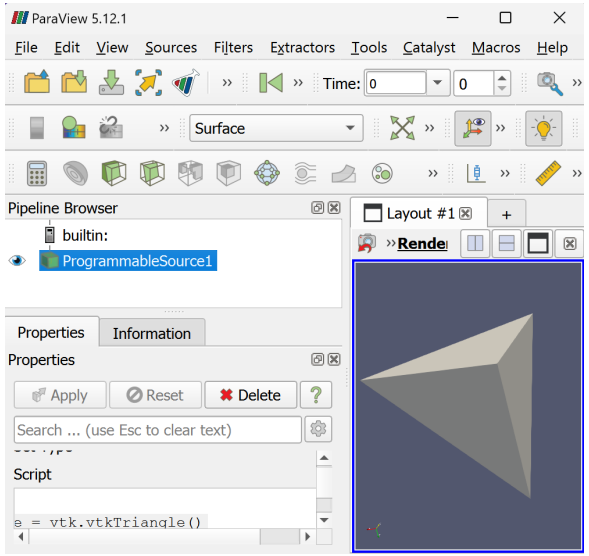
Python support not available here

PythonCalculator



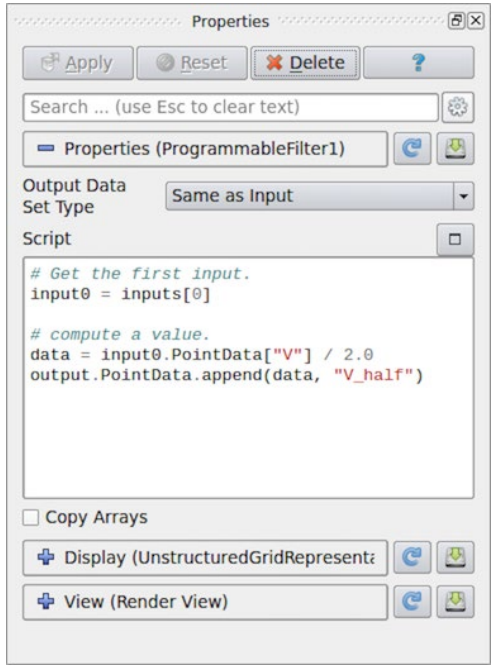
Calculator + support for Python based expressions

ProgrammableSource



generate custom data from scratch using Python scripting

ProgrammableFilter



enables users to manipulate and process existing data using Python scripts

demo

Example ProgrammableSource script:
workshop_2024/solution/programmable_source_example/script.txt
ParaView's Python documentation at:
https://kitware.github.io/paraview-docs/latest/python/paraview.simple.Calculator.html

sherin.sugathan@dscience.uio.no

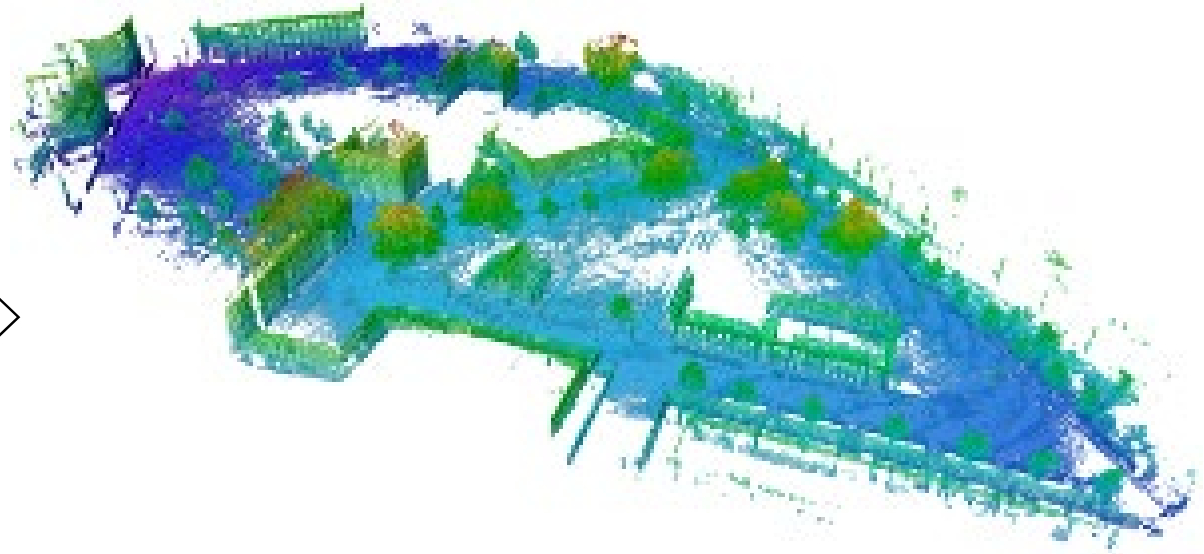
Python Scripting in ParaView

Loading your custom datasets

Task: Create a ProgrammableSource and load the provided custom dataset in ParaView

scan_001_points.dat

```
0.217 -0.074 0.713 -10.276 5.891 0.646
0.217 -0.074 0.713 -10.677 5.928 0.505
0.217 -0.074 0.713 -10.181 5.474 0.384
0.217 -0.074 0.713 -9.984 5.194 0.264
0.217 -0.074 0.713 -8.932 4.496 0.199
0.217 -0.074 0.713 -8.025 3.906 0.150
0.217 -0.074 0.713 -7.674 3.606 0.080
0.217 -0.074 0.713 -6.844 3.103 0.064
0.217 -0.074 0.713 -6.130 2.680 0.055
0.217 -0.074 0.713 -5.508 2.318 0.054
```



Relevant files are named:
scan_001_points.dat,
scan_002_points.dat,
scans_003_points.dat, …

Use dataset from location:
workshop_2024/exercise/custom_dataset/freiburgCampus360_3D
Script to be used inside ProgrammableSource:
workshop_2024/solution/programmable_source_example/script.txt
Original dataset:
recorded by B. Steder, University of Freiburg. (link)

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

What?

ParaView's functionality can be expanded with plugins. These plugins are compiled code libraries that can be loaded into ParaView as needed.

Extending ParaView with Plugins

Plugin Types

XML Plugins

Python Plugins

C++ Plugins

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

XML Plugins

When?

Expose a new VTK filter (because ParaView expose a subset of VTK)

Changing Existing VTK filter configuration to match your specific needs.

How to write an XML Plugin?

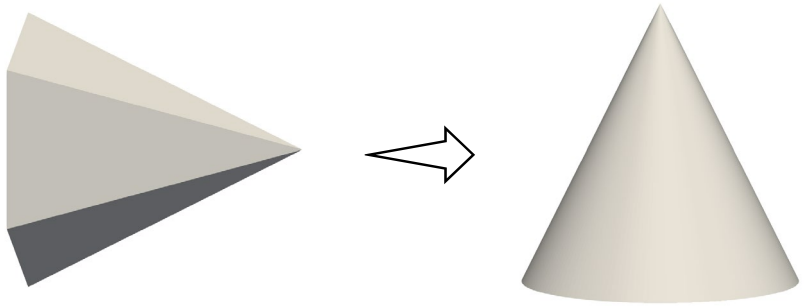
1. Copy paste existing readers/filters/sources/… from `Remoting/Application/Resources/*.xml`
2. Encapsulate in `<ServerManagerConfiguration>` and `<ProxyGroup>` elements.
3. Modify name and make other relevant changes in xml.
4. Load the plugin. (Tools → Manage Plugins…)

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

XML Plugins

Exercise



Change default cone resolution.

Change default cone orientation.

Remove unwanted properties.

demo

Refer ParaView source file:
Remoting/Application/Resources/sources_filterssources.xml
Use xml exercise file at:
workshop_2024/exercise/xml_plugin/MyCone.xml

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

Python Plugins

When?

Add your own computation script.

How to write a Python Plugin?

1. Use Python class inherited from `<VTKPythonAlgorithmBase>`
2. Implement `__init__()` and `RequestData()`
3. Advanced users can also implement

```
RequestInformation()
RequestUpdateExtent()
FillInputPortInformation(),
…
```

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

Python Plugins

```python
@smproxy.filter (name="MyPythonFilter" )
@smproperty.input (name="Input", port_index ="0")
@smdomain.datatype (dataTypes =["vtkPointSet"])
class MyPythonFilter (VTKPythonAlgorithmBase ):
    def __init__ (self):
        VTKPythonAlgorithmBase. __init__ (self,
        nInputPorts =1, nOutputPorts =1, outputType ='vtkPolyData')

    @smproperty.doublevector (name="Thickness" ,default_values =0.3)
    @smdomain.doublerange (min=1e-24, max=1.0)
    def SetThickness (self, x):
        self.Thickness = x
        self.Modified()

    def RequestData (self, request, inInfo, outInfo):
        # do work
        return 1
```

More Examples at:
https://gitlab.kitware.com/paraview/paraview/-/blob/master/Examples/Plugins/PythonAlgorithm/PythonAlgorithmExamples.py

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

Python Plugins

```
@smproxy.source
@smproxy.filter
@smproxy.reader
@smproxy.writer
```

```python
@smproxy.filter (name="MyPythonFilter" )
@smproperty.input (name="Input", port_index ="0")
@smdomain.datatype (dataTypes =["vtkPointSet"])
class MyPythonFilter (VTKPythonAlgorithmBase ):
    def __init__ (self):
        VTKPythonAlgorithmBase. __init__ (self,
        nInputPorts =1, nOutputPorts =1, outputType ='vtkPolyData')

    @smproperty.doublevector (name="Thickness" ,default_values =0.3)
    @smdomain.doublerange (min=1e-24, max=1.0)
    def SetThickness (self, x):
        self.Thickness = x
        self.Modified()

    def RequestData (self, request, inInfo, outInfo):
        # do work
        return 1
```

More Examples at:
https://gitlab.kitware.com/paraview/paraview/-/blob/master/Examples/Plugins/PythonAlgorithm/PythonAlgorithmExamples.py

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

Python Plugins

```
@smproperty.xml
@smproperty.intvector
@smproperty.doublevector
@smproperty.stringvector
@smproperty.proxy
@smproperty.input
@smproperty.dataarrayselection
```

```python
@smproxy.filter (name="MyPythonFilter" )
@smproperty.input (name="Input", port_index ="0")
@smdomain.datatype (dataTypes =["vtkPointSet"])
class MyPythonFilter (VTKPythonAlgorithmBase ):
    def __init__ (self):
        VTKPythonAlgorithmBase. __init__ (self,
        nInputPorts =1, nOutputPorts =1, outputType ='vtkPolyData')

    @smproperty.doublevector (name="Thickness" ,default_values =0.3)
    @smdomain.doublerange (min=1e-24, max=1.0)
    def SetThickness (self, x):
        self.Thickness = x
        self.Modified()

    def RequestData (self, request, inInfo, outInfo):
        # do work
        return 1
```

More Examples at:
https://gitlab.kitware.com/paraview/paraview/-/blob/master/Examples/Plugins/PythonAlgorithm/PythonAlgorithmExamples.py

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

Python Plugins

```
@smdomain.xml
@smdomain.intrange
@smdomain.doublerange
@smdomain.filelist
@smdomain.datatype
```

```python
@smproxy.filter (name="MyPythonFilter" )
@smproperty.input (name="Input", port_index ="0")
@smdomain.datatype (dataTypes =["vtkPointSet"])
class MyPythonFilter (VTKPythonAlgorithmBase ):
    def __init__ (self):
        VTKPythonAlgorithmBase. __init__ (self,
        nInputPorts =1, nOutputPorts =1, outputType ='vtkPolyData')

    @smproperty.doublevector (name="Thickness" ,default_values =0.3)
    @smdomain.doublerange (min=1e-24, max=1.0)
    def SetThickness (self, x):
        self.Thickness = x
        self.Modified()

    def RequestData (self, request, inInfo, outInfo):
        # do work
        return 1
```

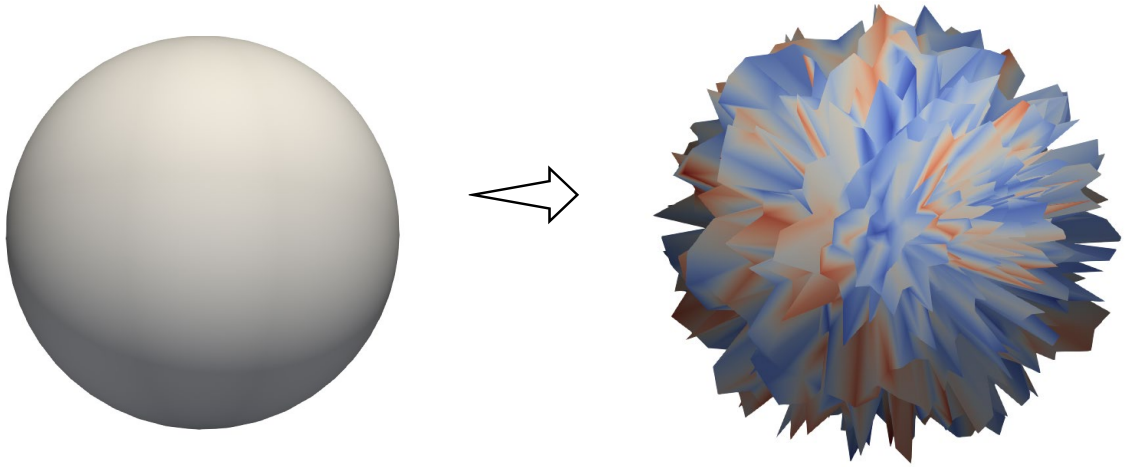
More Examples at:
https://gitlab.kitware.com/paraview/paraview/-/blob/master/Examples/Plugins/PythonAlgorithm/PythonAlgorithmExamples.py

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

Python Plugins

Exercise



Apply random distortion to a sphere

Use source file from location:
workshop_2024/exercise_data/python_plugin/MyRandomWarp.py
More Examples at:
https://gitlab.kitware.com/paraview/paraview/-/blob/master/Examples/Plugins/PythonAlgorithm/PythonAlgorithmExamples.py

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

C++ Plugins

XML Plugins are simple to use but can only use existing VTK/ParaView filters. You cannot add your own computation scripts directly.

Python Plugins allows you to write your own Python code as in a Programmable filter. Automatically generated GUI.
Not geared towards performance.

C++ Plugins provides more flexibility and performance. Can directly access VTK/ParaView internals. Manual GUI development.
Requires to know C++ and you need to build ParaView

sherin.sugathan@dscience.uio.no

Extending ParaView with Plugins

C++ Plugins

demo

Detailed documentation on writing ParaView plugins available at

https://www.paraview.org/Wiki/ParaView/Plugin_HowTo

sherin.sugathan@dscience.uio.no

Python Scripting in ParaView

Using virtual environments inside ParaView (pvpython)

Setting up your environment

```
> python -m virtualenv venv
> myvenv\Scripts\activate
> pip install numpy
```

Testing Inside ParaView

```
>>> activate_this = "C:/pv/workshop_2024/custom_envs/myvenv/Scripts/activate_this.py"
>>> exec(open(activate_this).read(), {'__file__': activate_this})
>>> import numpy as np
```

demo

sherin.sugathan@dscience.uio.no

ParaView in HPC Environments

Visualization Approaches

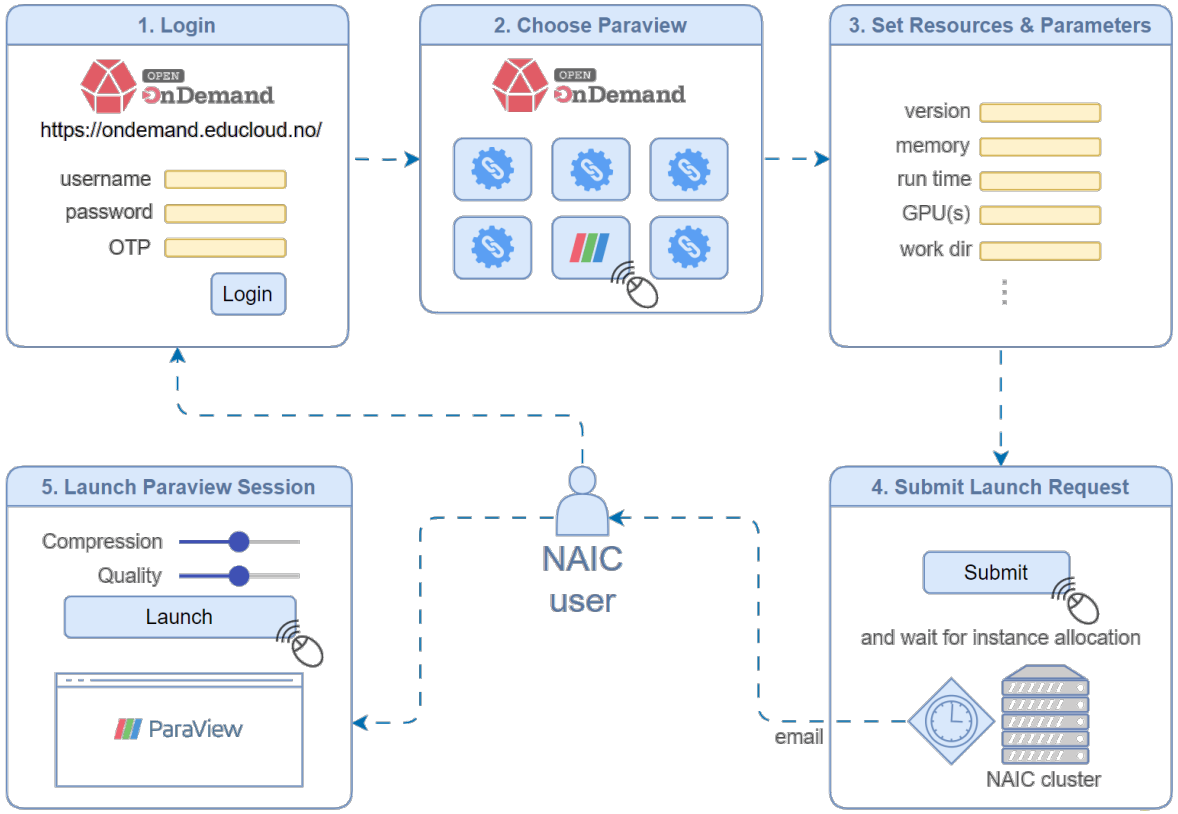
Batch based Visualization

In-situ Visualization

Exploratory Visualization

ParaView in HPC Environments

How? - User Workflow Under NAIC



sherin.sugathan@dscience.uio.no

ParaView in HPC Environments

Using EduCloud OnDemand

1

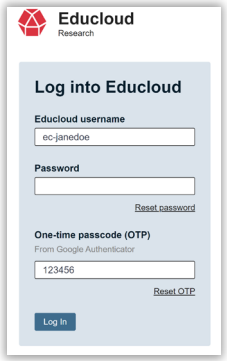
Login

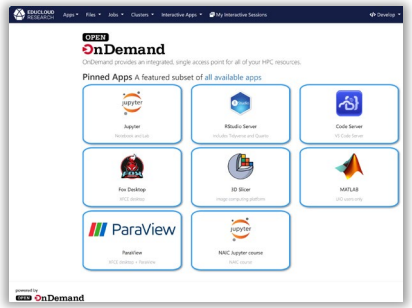
2

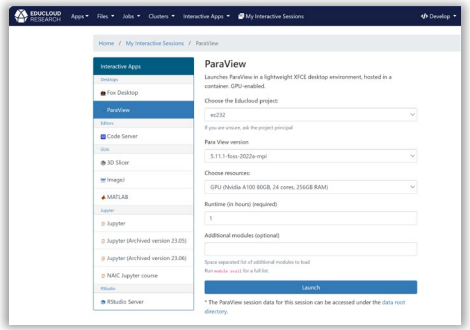
Request

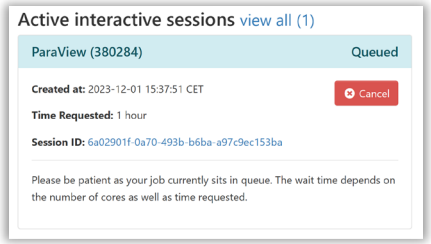
3

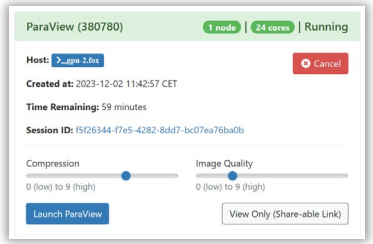
Use

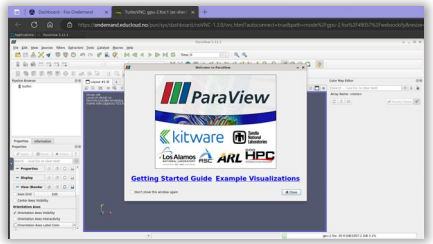








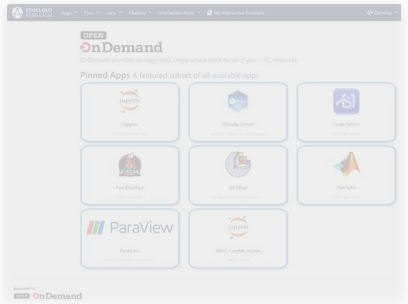




sherin.sugathan@dscience.uio.no

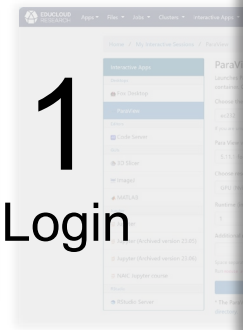
ParaView in HPC Environments

Using EduCloud OnDemand



1

Login



Educloud
Research

Log into Educloud

Educloud username

ec-janedoe

Password

Reset password

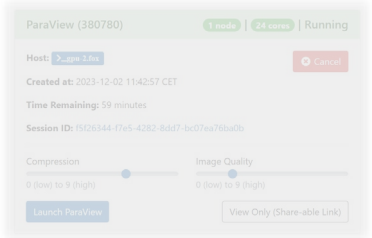
One-time passcode (OTP)

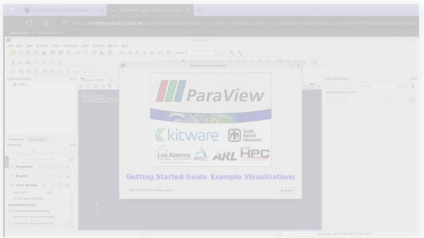
From Google Authenticator

123456

Reset OTP

Log In





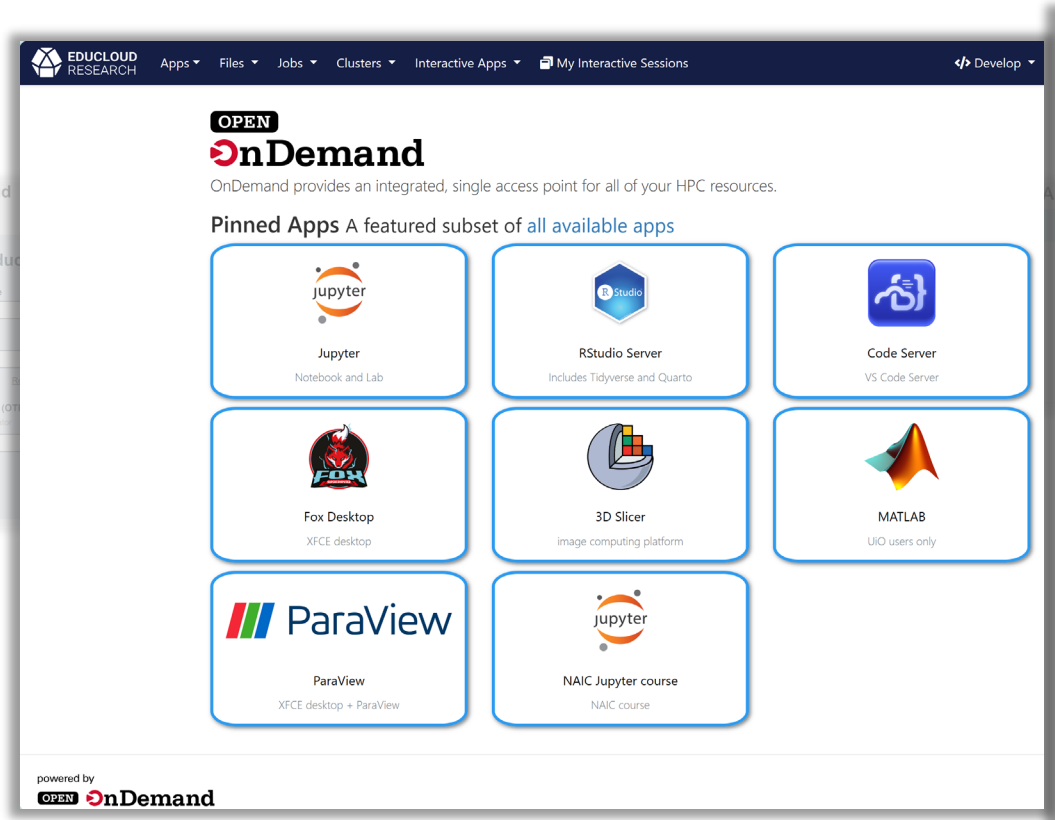
sherin.sugathan@dscience.uio.no

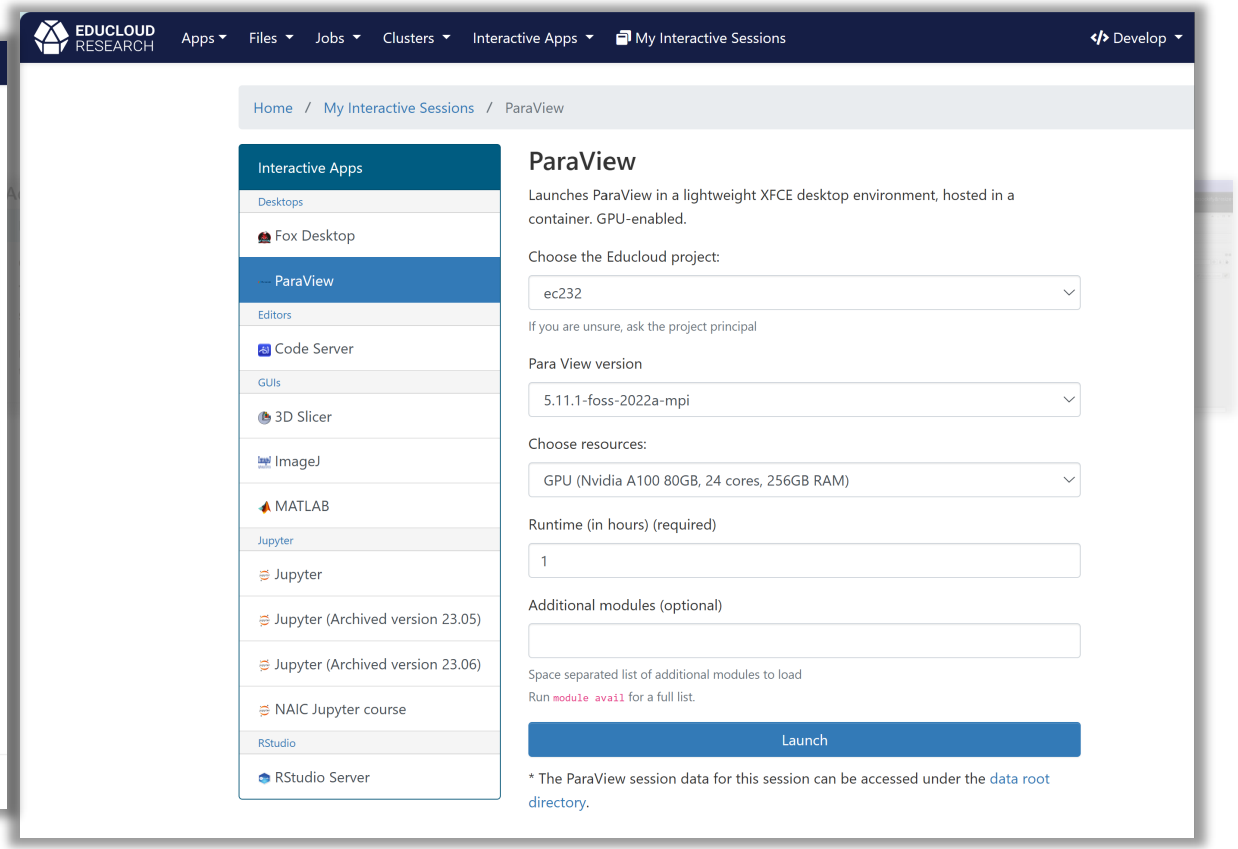
ParaView in HPC Environments

Using EduCloud OnDemand

2

Request

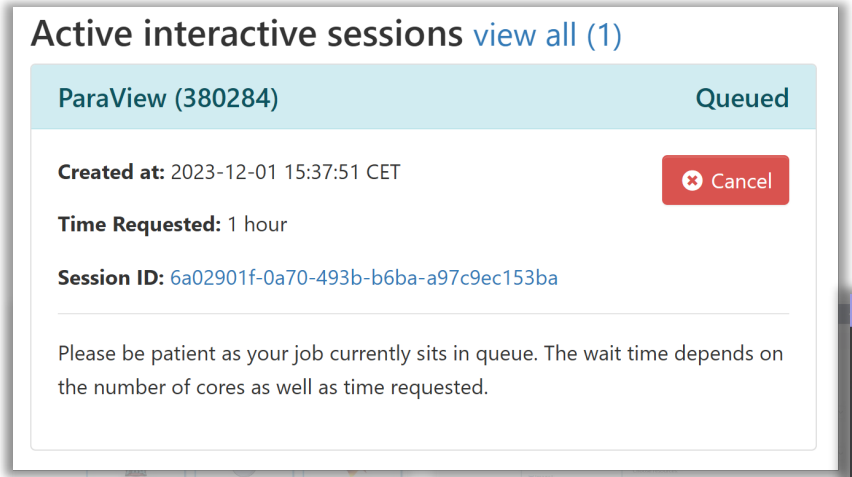




sherin.sugathan@dscience.uio.no

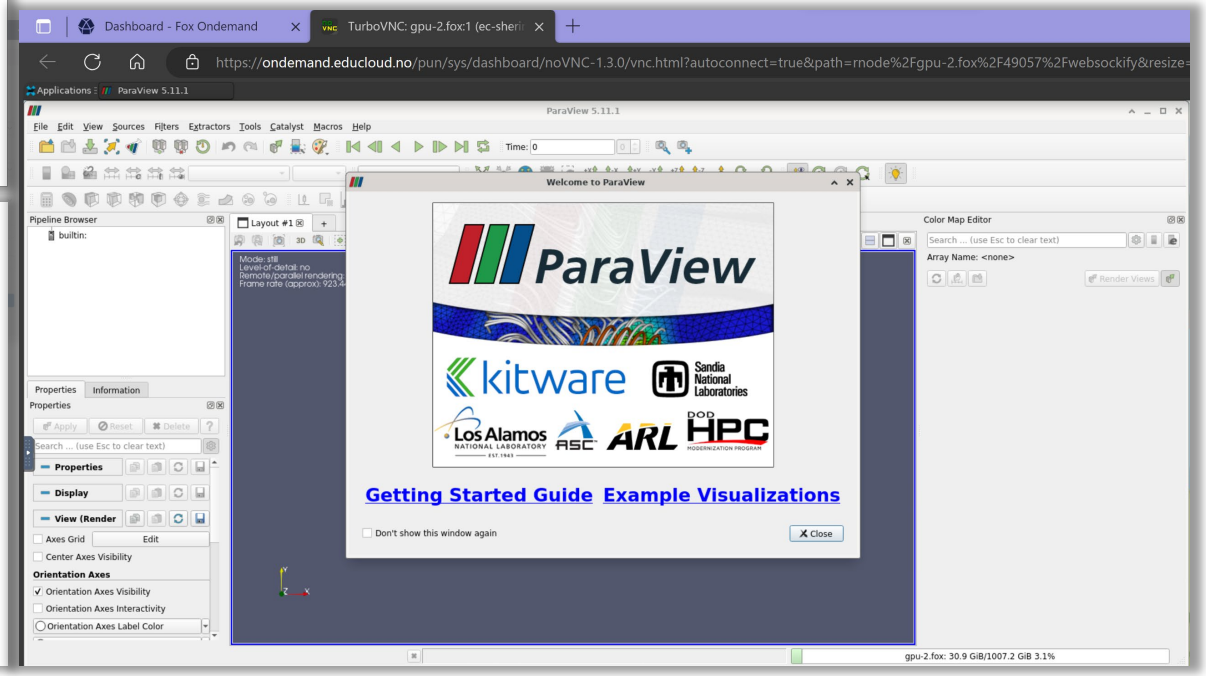
ParaView in HPC Environments

Using EduCloud OnDemand



3

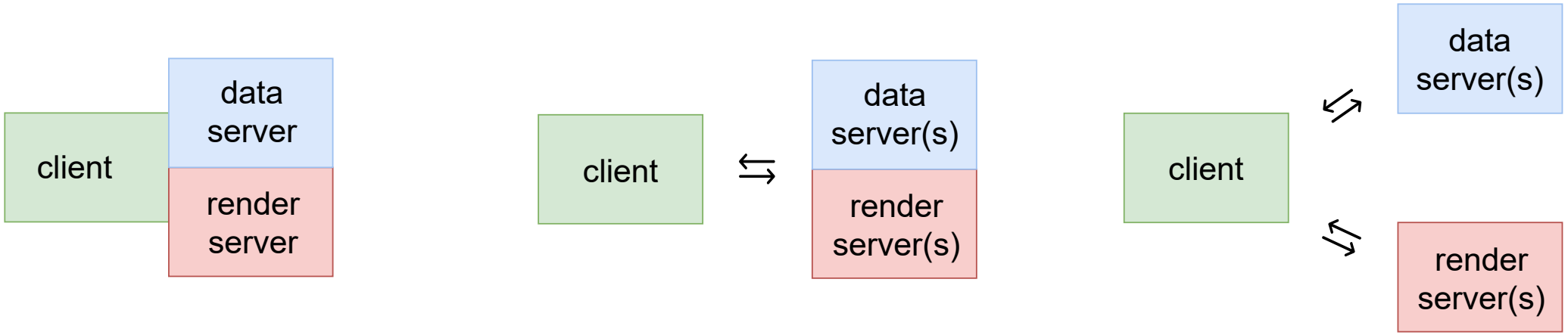
Use



sherin.sugathan@dscience.uio.no

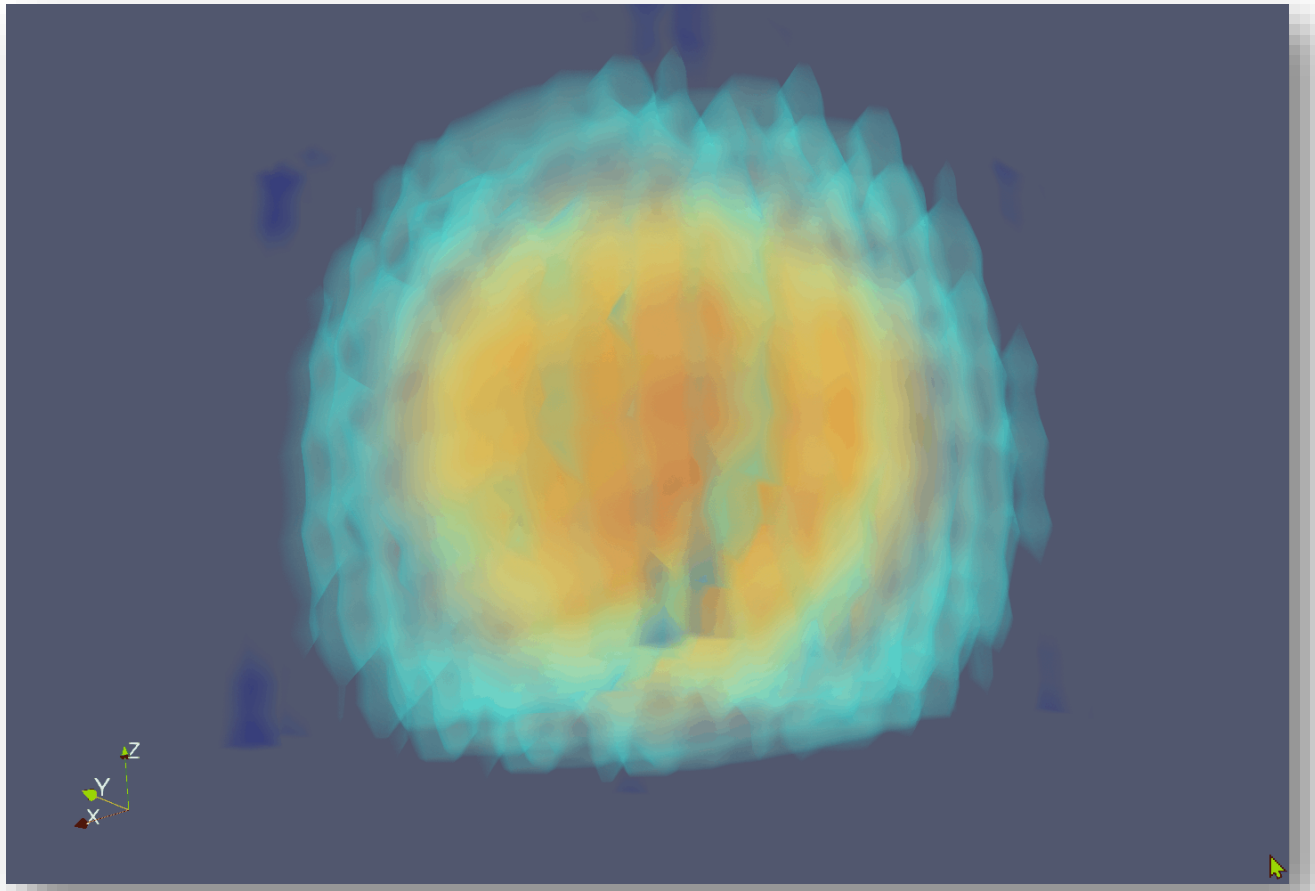
ParaView

Client Server Architecture



Local vs HPC

Wavelet Visualization

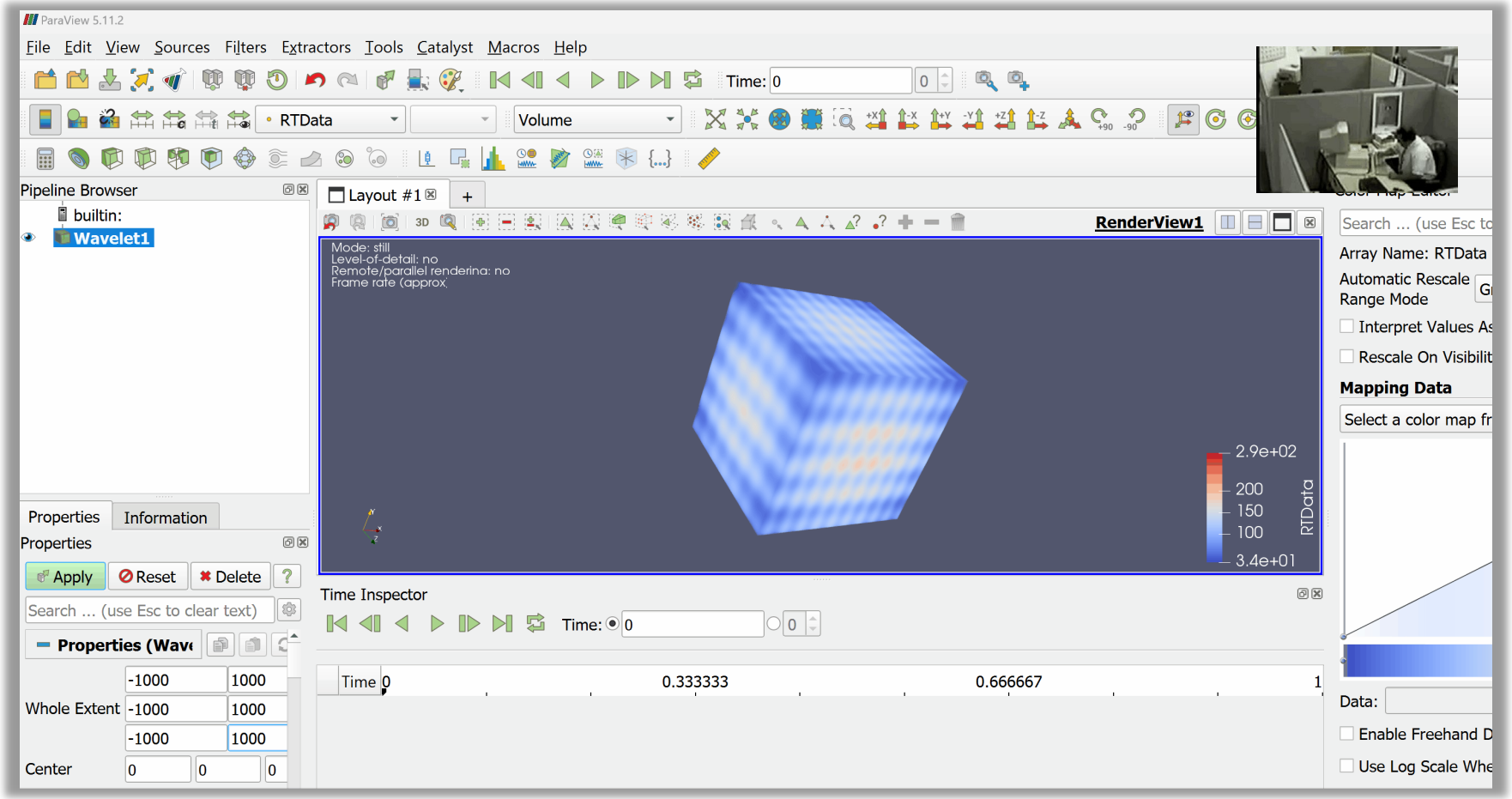


	-10	10	
Whole Extent	-10	10	
	-10	10	
Center	0	0	0

sherin.sugathan@dscience.uio.no

Local vs HPC

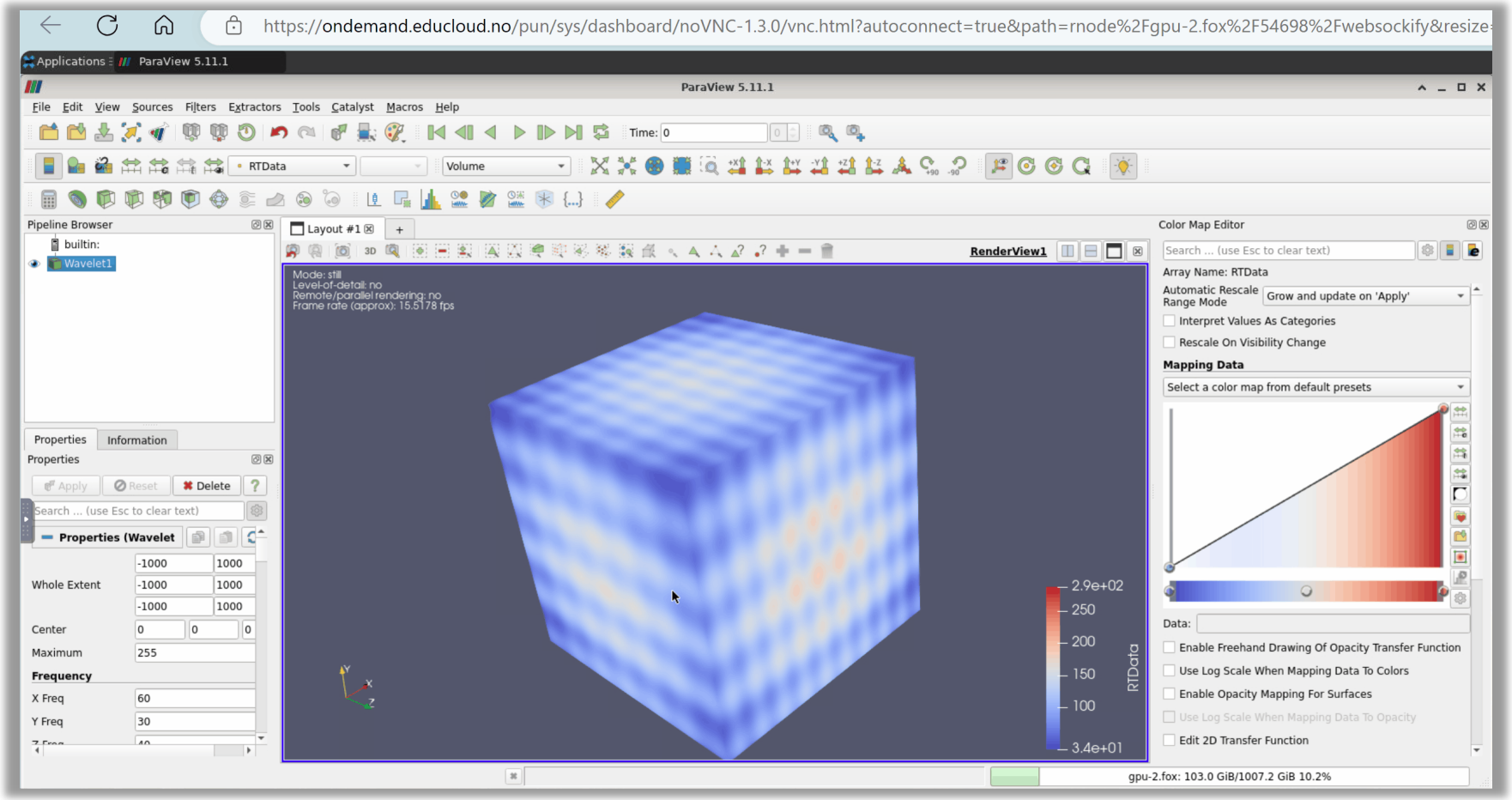
Wavelet Visualization



sherin.sugathan@dscience.uio.no

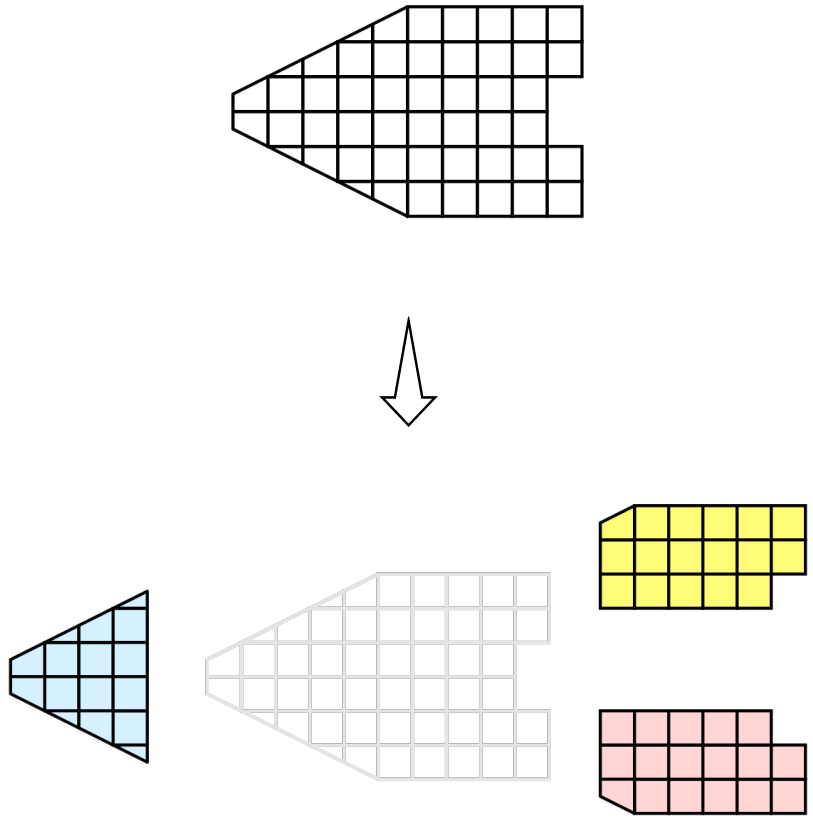
Local vs HPC

Wavelet Visualization



Parallel Processing

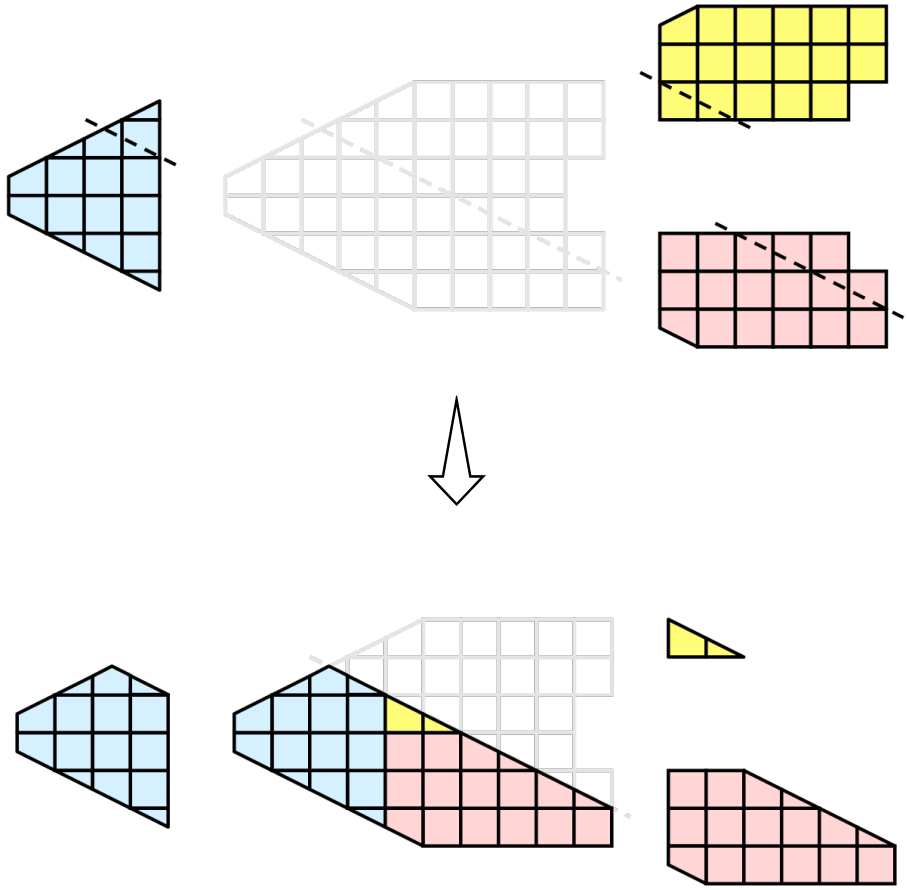
Divide and Conquer



Images from:
https://docs.paraview.org

Parallel Processing

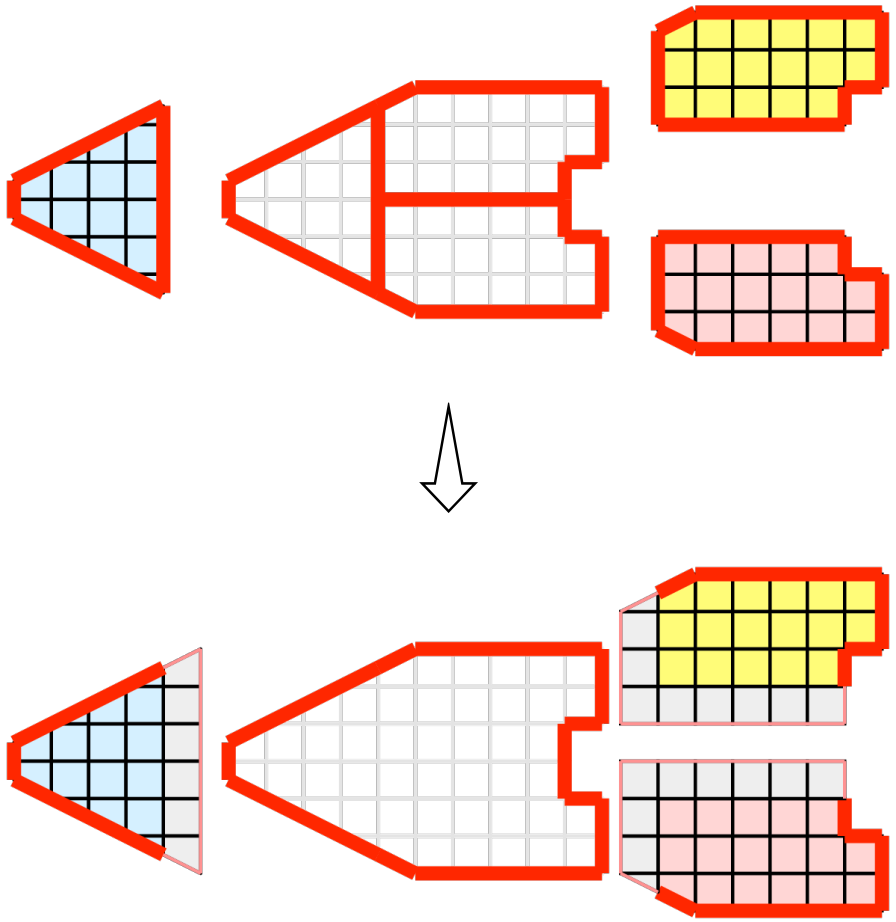
Example: Clipping a dataset in parallel



Images from:
https://docs.paraview.org

Parallel Processing

The need for Ghost Cells



Images from:
https://docs.paraview.org

MPI

Message Passing Interface

The Message Passing Interface (MPI) is an Application Program Interface that defines a model of parallel computing where each parallel process has its own local memory, and data must be explicitly shared by passing messages between processes.

An MPI parallel code requires some changes from serial code, as MPI function calls to communicate data are added, and the data must somehow be divided across processes.

Text from:
https://carleton.ca/rcs/rcdc/introduction-to-mpi/

Using Parallel Capabilities of ParaView

Establishing Connections – ML Nodes

1. You can use the following command to move your large datasets to a remote location.

```
scp large_dataset.vtk user@remote-server:/path/path
```

2. Connect to remote server, load ParaView module, start pvserver using the following command

```
mpirun -np 6 pvserver -server-port=5556
```

```
Waiting for client...
Connection URL: cs://ml3.hpc.uio.no:5556
Accepting connection(s): ml3.hpc.uio.no:5556
```

3. Open a local terminal, set up a ssh tunnel from your client to the server.

```
ssh -J sherinsu@login.uio.no -L 5556:localhost:5556 sherinsu@ml3.hpc.uio.no
```

4. Start your local ParaView GUI instance and connect to remote server using the following settings.

```
Server Type: Client/Server     Host: localhost     Port:5556
```

Note: ML Nodes are intended for machine learning and deep learning tasks. No batch/queue system. Use the machines in a solidaric way.
Please learn about resource details and usage guidelines from:
https://www.uio.no/tjenester/it/forskning/kompetansehuber/uio-ai-hub-node-project/it-resources/ml-nodes/

sherin.sugathan@dscience.uio.no

Using Parallel Capabilities of ParaView

Establishing Connections – EduCloud OnDemand

1. You can use the following command to move your large datasets to a project storage location under EduCloud.

```
scp large_dataset.vtk user@remote-server:/path/path
```

2. Connect to Fox EduCloud, and submit a job to start pvserver using MPI (check workshop data folder for full script (example))

```
mpirun --map-by ppr:1:core --bind-to core -np 50 pvserver --server-port=5556
```

```
Waiting for client...
Connection URL: cs://c1-8.fox:5556
Accepting connection(s): c1-8.fox:5556
```

3. Start a ParaView instance using EduCloud OnDemand.

4. Connect to remote server using the following settings.

```
Server Type: Client/Server     Host: c1-8.fox     Port:5556
```

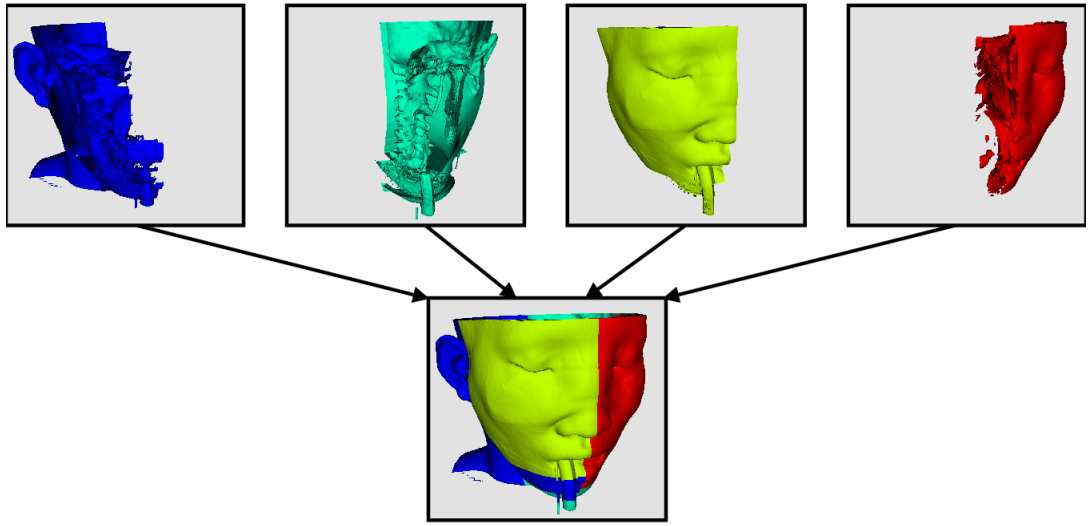
Note: The commands and procedures for using ParaView under EduCloud OnDemand may change over time as the setup is improved. Always refer to the latest methods provided here.

Sample slurm scripts at:
workshop_2024/solution/hpc

sherin.sugathan@dscience.uio.no

Parallel Rendering

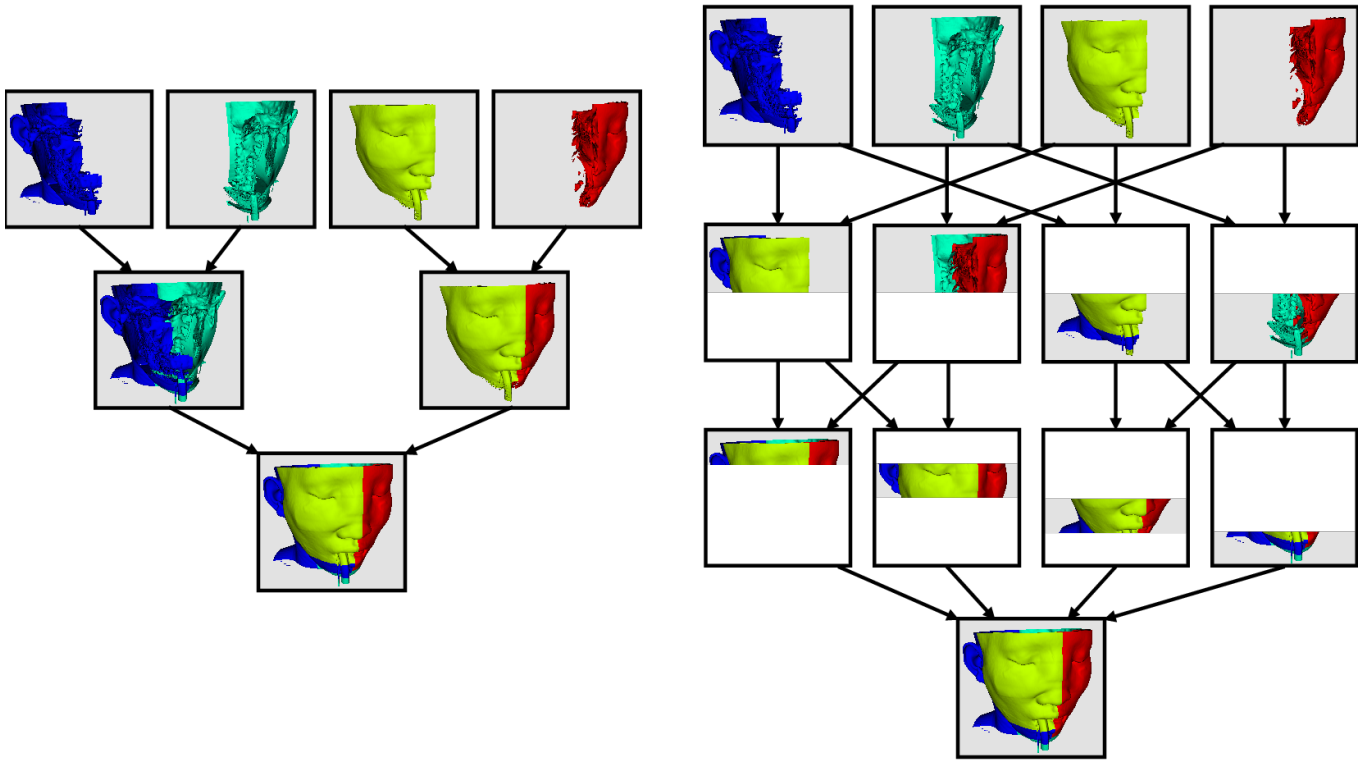
Uses IceT



Images from:
https://docs.paraview.org

Parallel Rendering

Image Compositing Algorithms

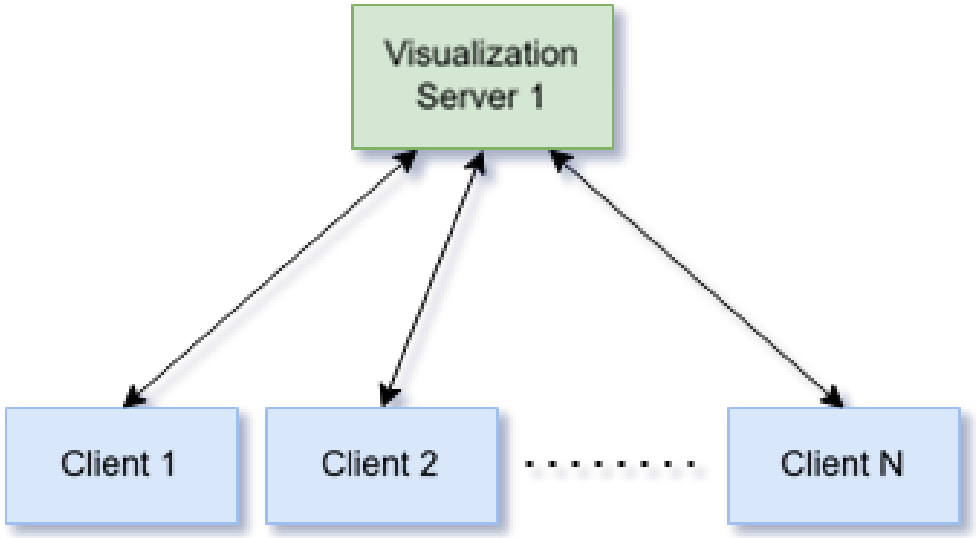


More documentation available at:
https://www.sandia.gov/app/uploads/sites/150/2021/10/IceTUsersGuide-2-0.pdf

sherin.sugathan@dscience.uio.no

Collaborative Visualization using HPC

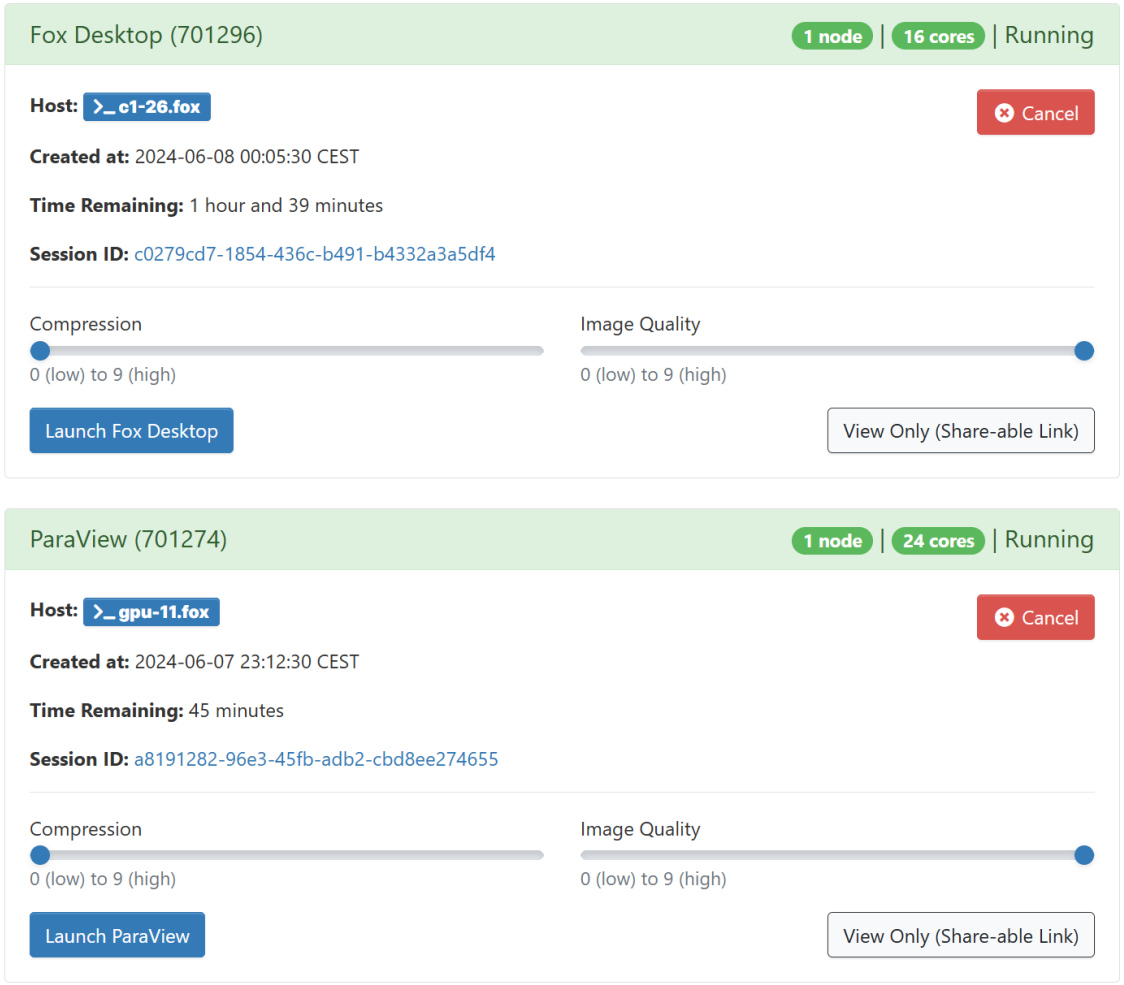
One Server, Multiple Clients



sherin.sugathan@dscience.uio.no

Collaborative Visualization using HPC

One Server, Multiple Clients

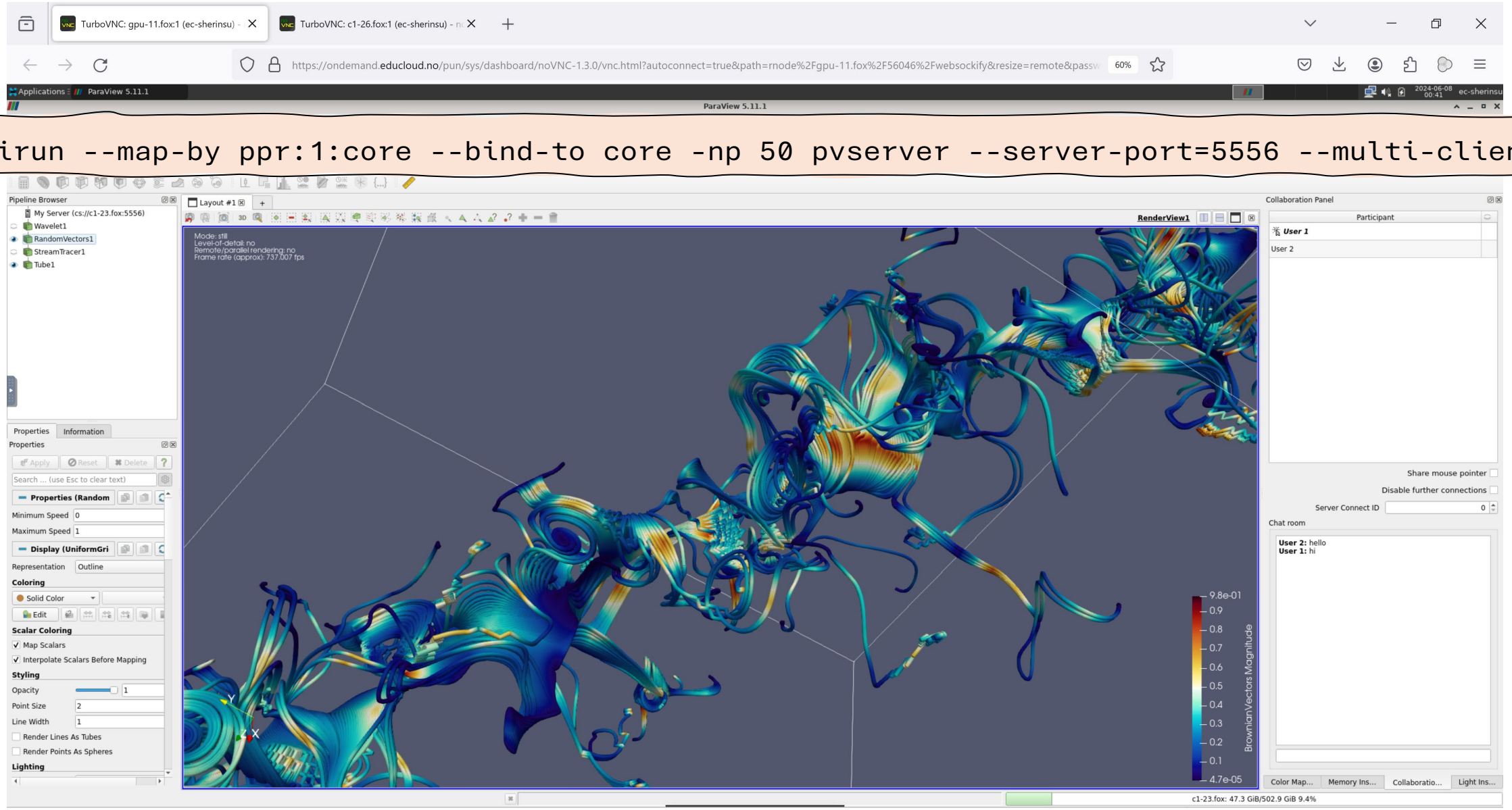


sherin.sugathan@dscience.uio.no

Collaborative Visualization using HPC

One Server, Multiple Clients

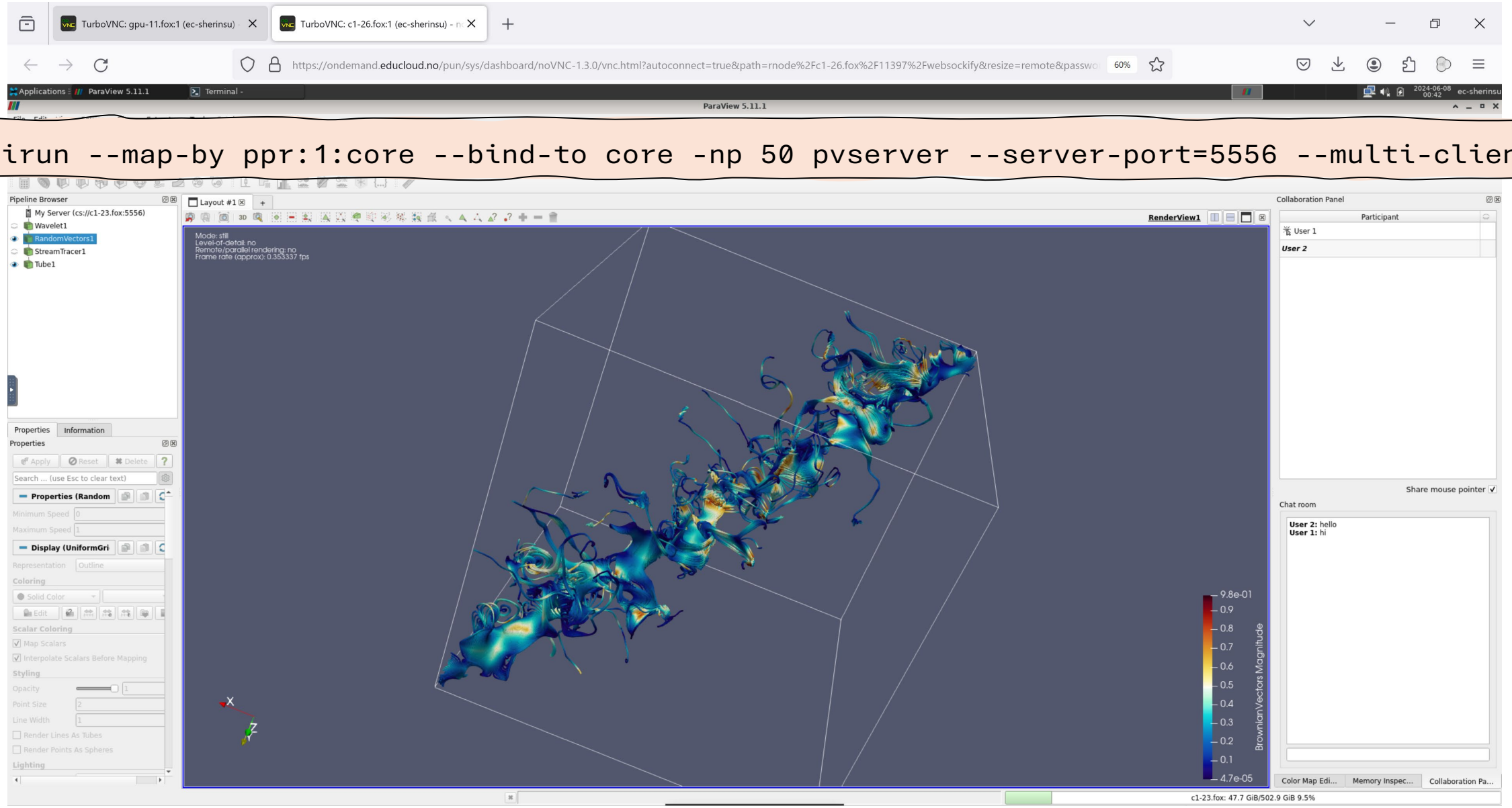
```
mpirun --map-by ppr:1:core --bind-to core -np 50 pvserver --server-port=5556 --multi-clients
```



Collaborative Visualization using HPC

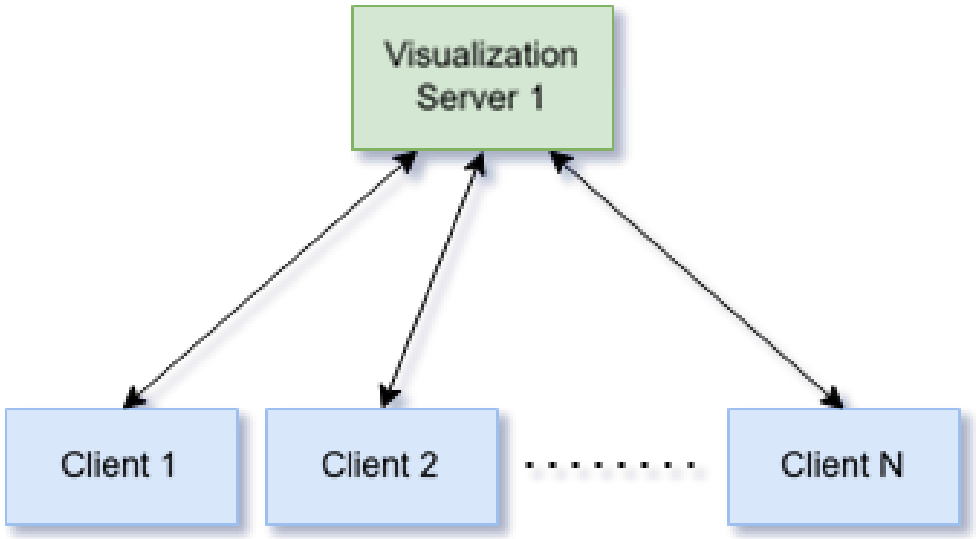
One Server, Multiple Clients

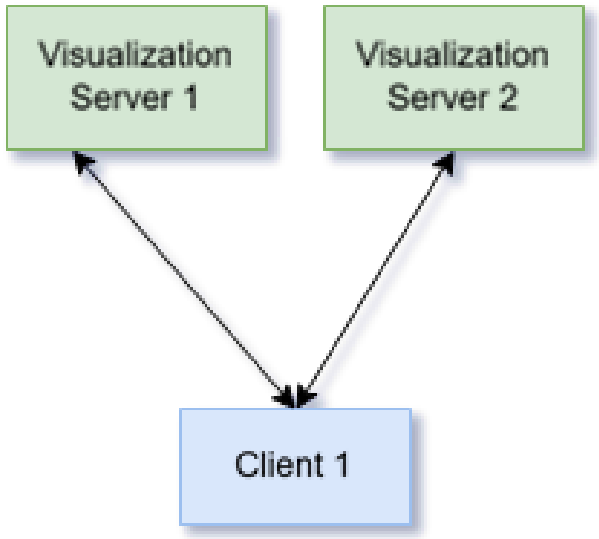
```
mpirun --map-by ppr:1:core --bind-to core -np 50 pvserver --server-port=5556 --multi-clients
```



Collaborative Visualization using HPC

One Client, Multiple Servers





sherin.sugathan@dscience.uio.no

Incorporating AI Pipelines in ParaView

Deep Learning for Object Detection

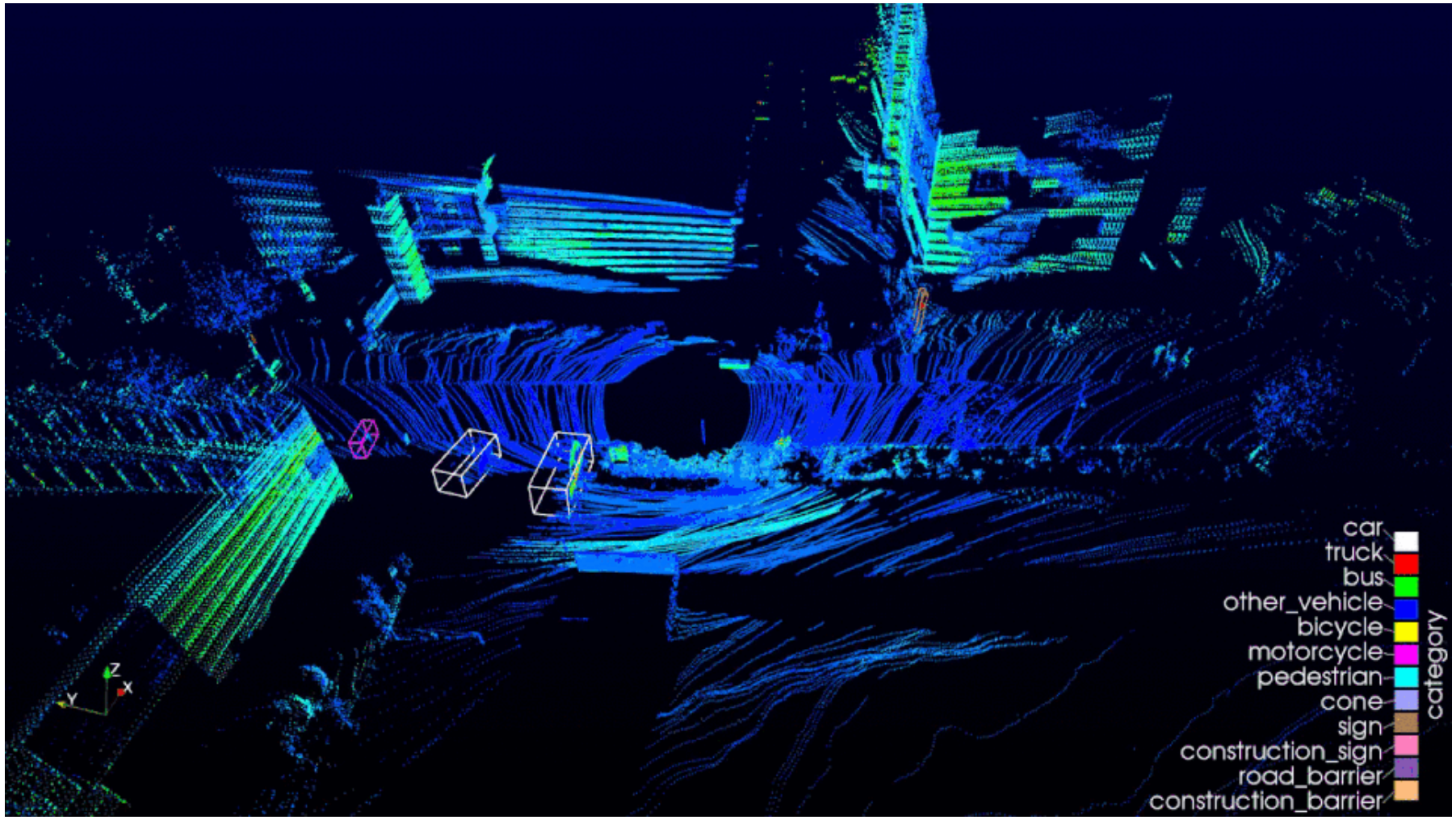
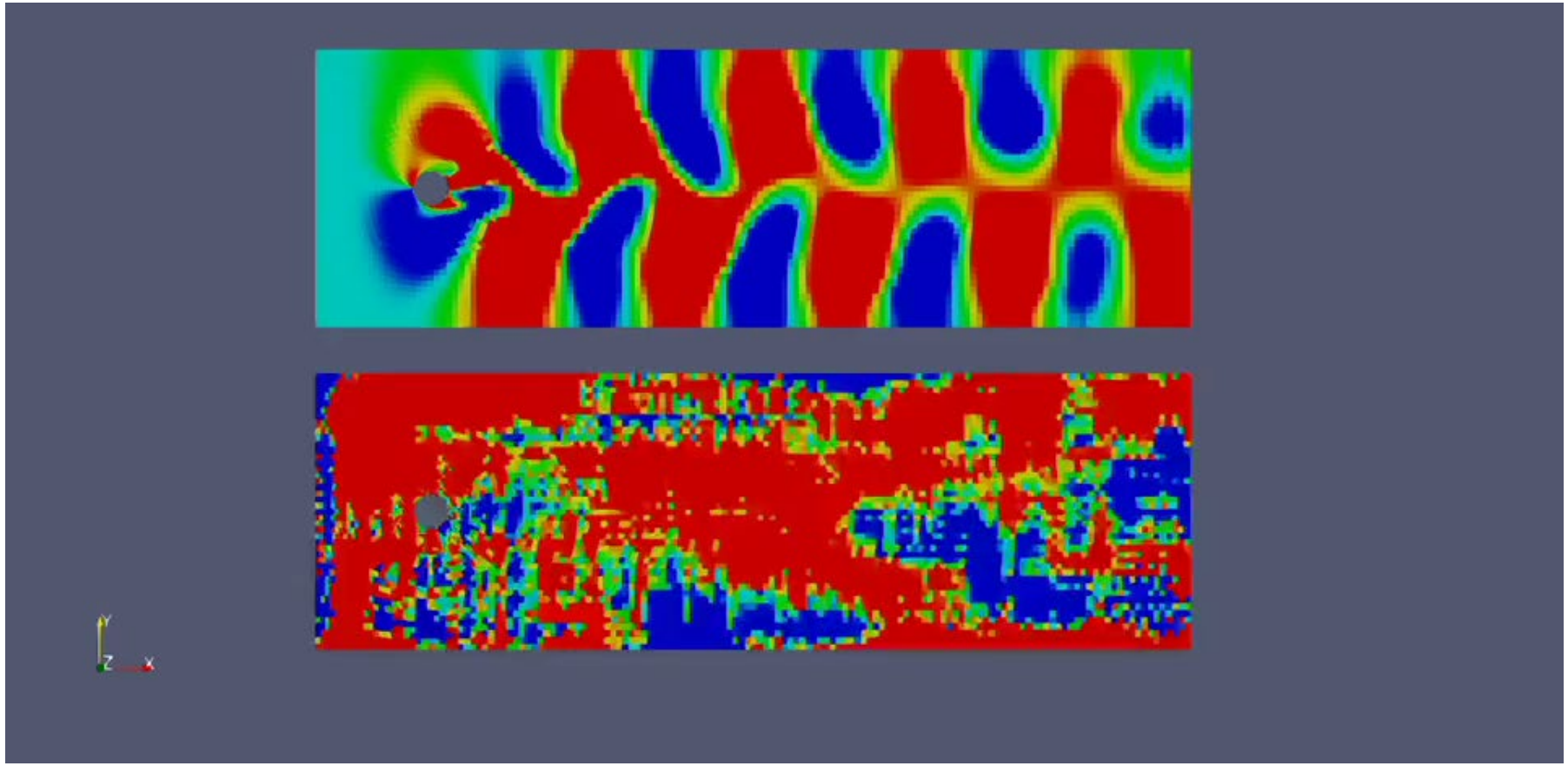


Image:
https://www.kitware.com/ai-point-cloud-deep-learning-in-paraview-and-lidarview/

sherin.sugathan@dscience.uio.no

Incorporating AI Pipelines in ParaView

Monitoring the Training Process

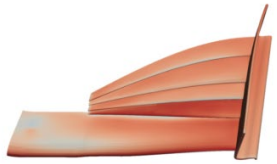


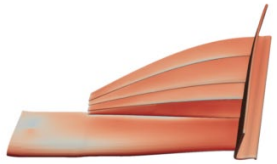
Reference:
Meyer, Lucas, et al. "Deep Surrogate for Direct Time Fluid Dynamics." *arXiv preprint arXiv:2112.10296* (2021).

sherin.sugathan@dscience.uio.no

Incorporating AI Pipelines in ParaView

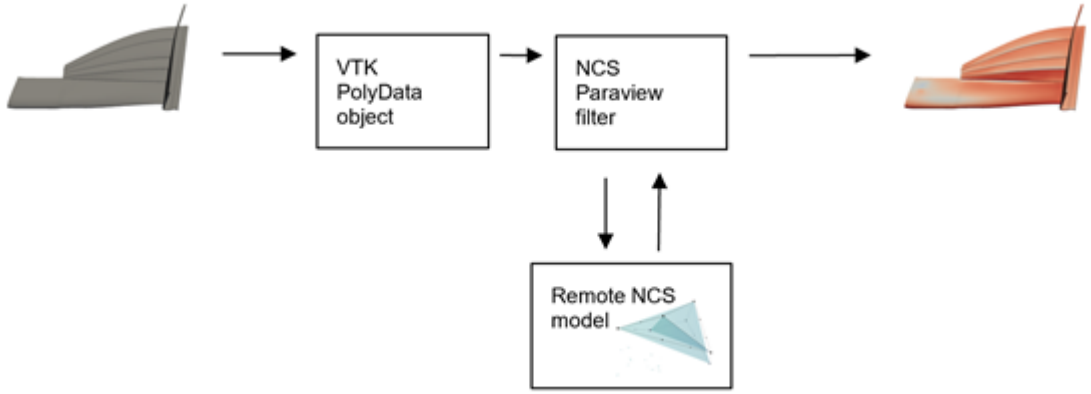
Geometric Deep-Learning Models





CFD
simulation: 2
hours

NCS
prediction: 50
ms

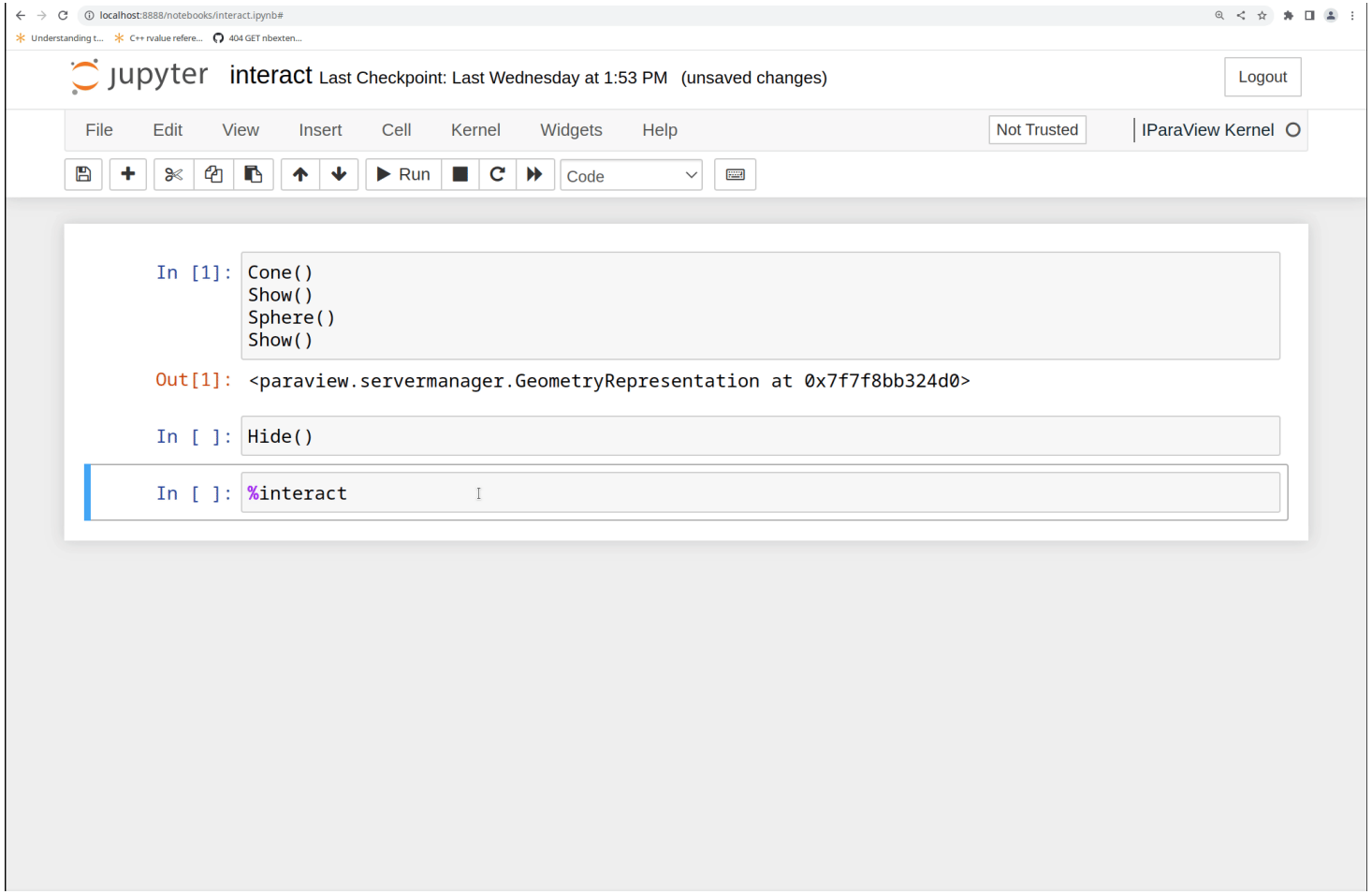


Plugin written by:
Neural Concept (https://neuralconcept.com/)
Images from:
https://www.kitware.com/integrating-geometric-deep-learning-models-into-paraview/

sherin.sugathan@dscience.uio.no

Other Interfaces

Controlling ParaView from Jupyter Notebook



IParaView Jupyter Kernel:
https://gitlab.kitware.com/paraview/plugins/iparaview-kernel

Other Interfaces

Custom Web-based User Interfaces for ParaView

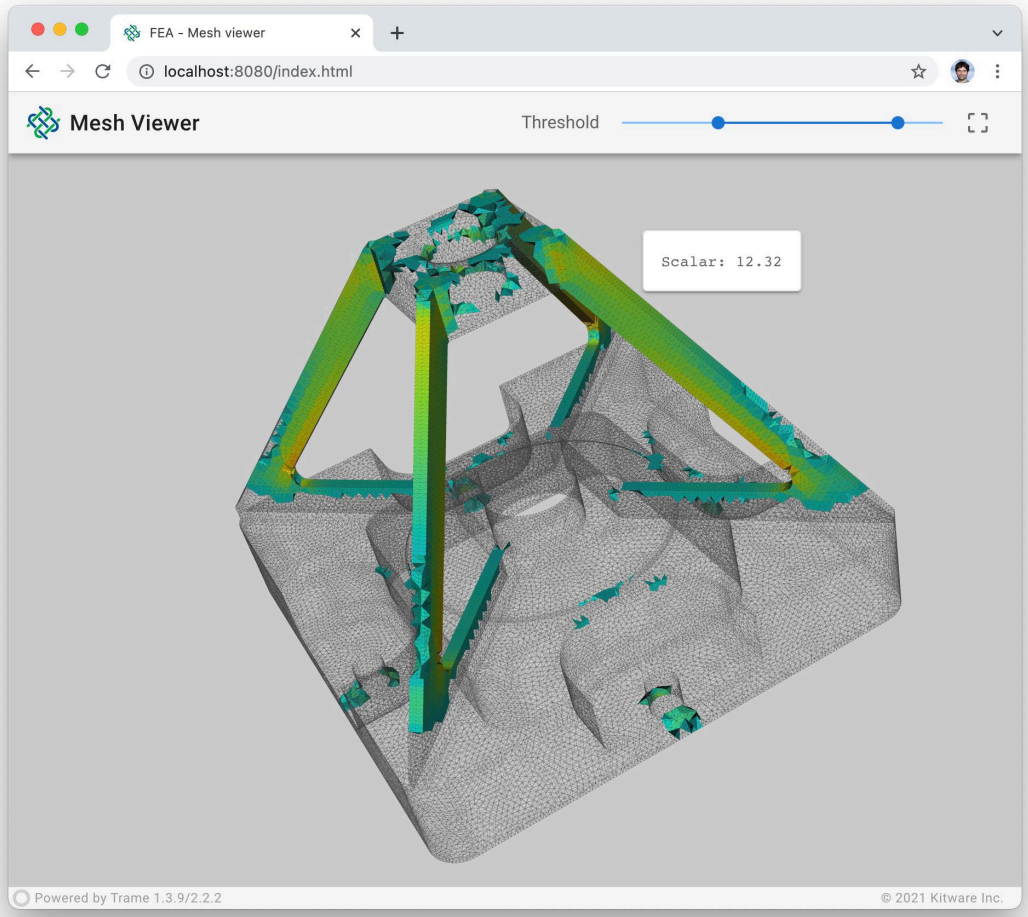


Image from:
Kitware's vtk-examples
Try out the source and commands at:
workshop_2024/solution/trame

sherin.sugathan@dscience.uio.no

Expanding ParaView's Capabilities

AR/VR/XR

AR/VR/XR

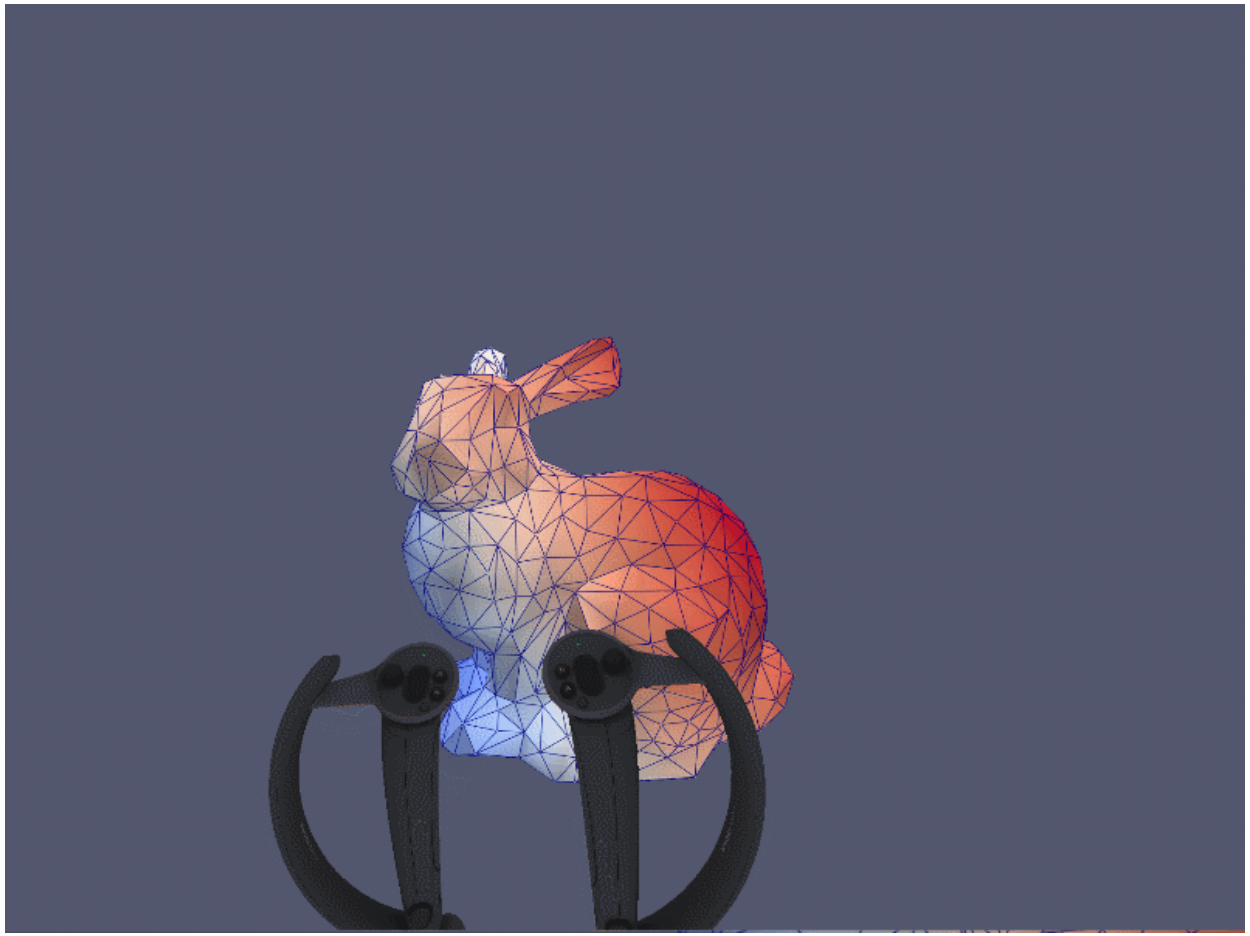


Image taken from:
https://www.kitware.com/navigation-basics-in-virtual-reality-with-paraview/

sherin.sugathan@dscience.uio.no

Expanding ParaView's Capabilities

Integration with Omniverse, XR, and Tiled Displays

NVIDIA Omniverse

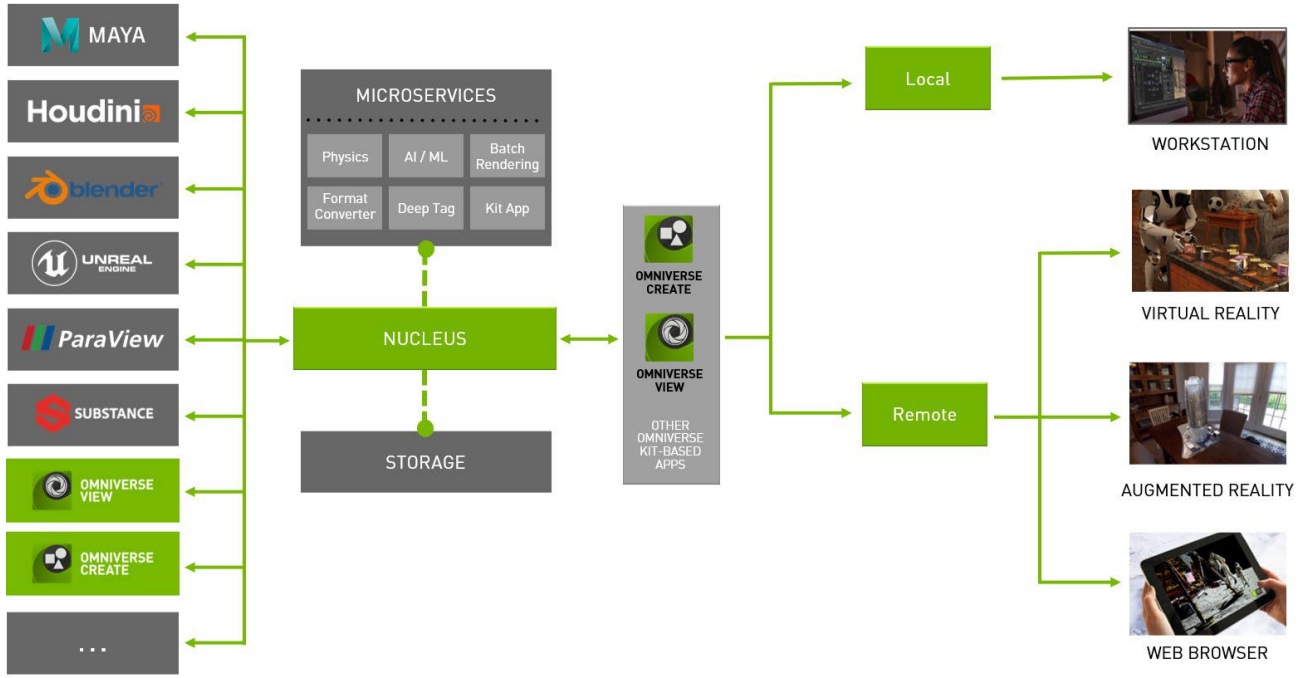


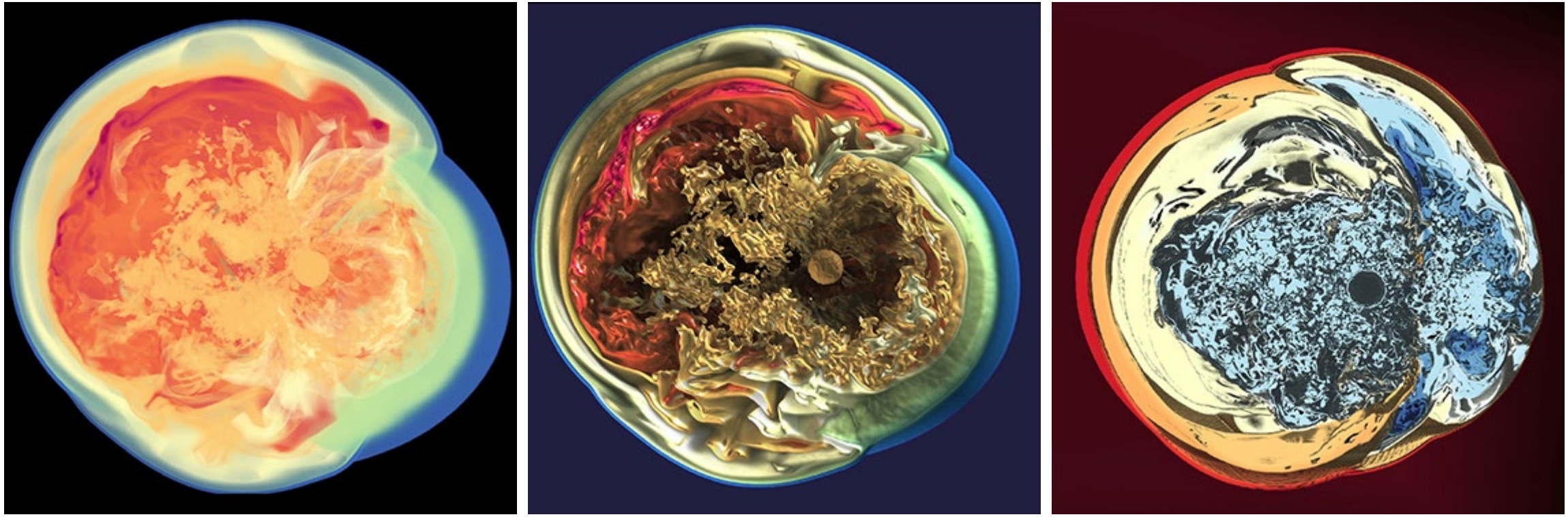
Image taken from:
https://www.kitware.com/navigation-basics-in-virtual-reality-with-paraview/
Reference:
https://www.nvidia.com/en-us/omniverse/

sherin.sugathan@dscience.uio.no

Advanced Rendering

NVIDIA Index

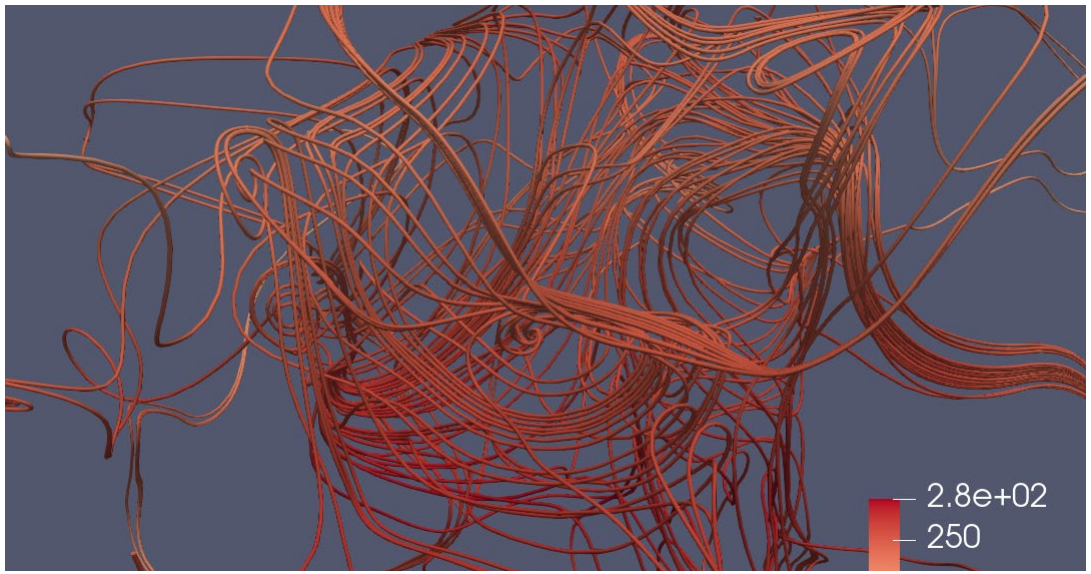
Core Collapse Supernova

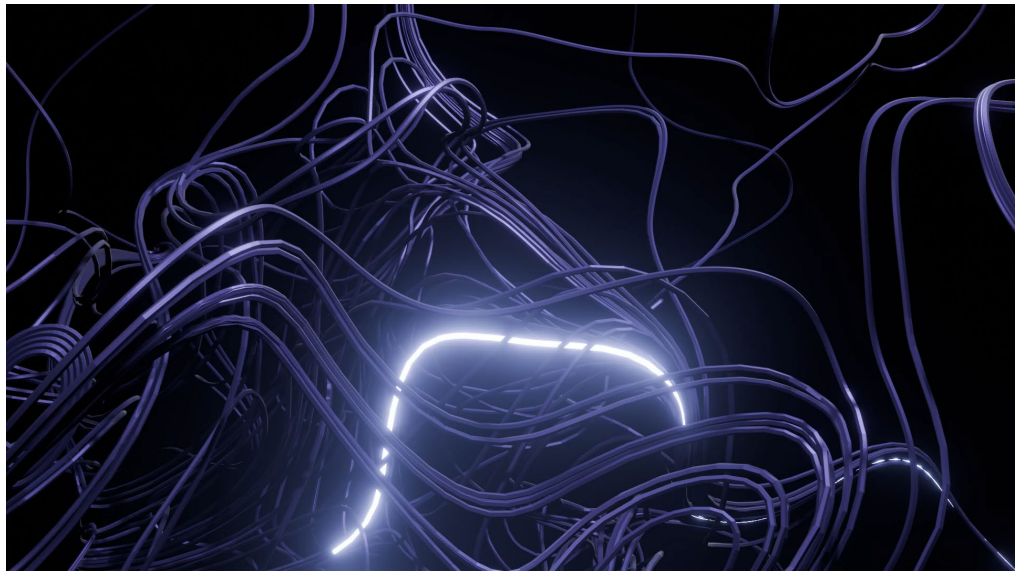


Images from:
https://www.nvidia.com/en-us/data-center/index-paraview-plugin/

Advanced Rendering

Exporting Data from ParaView for use in Other Tools



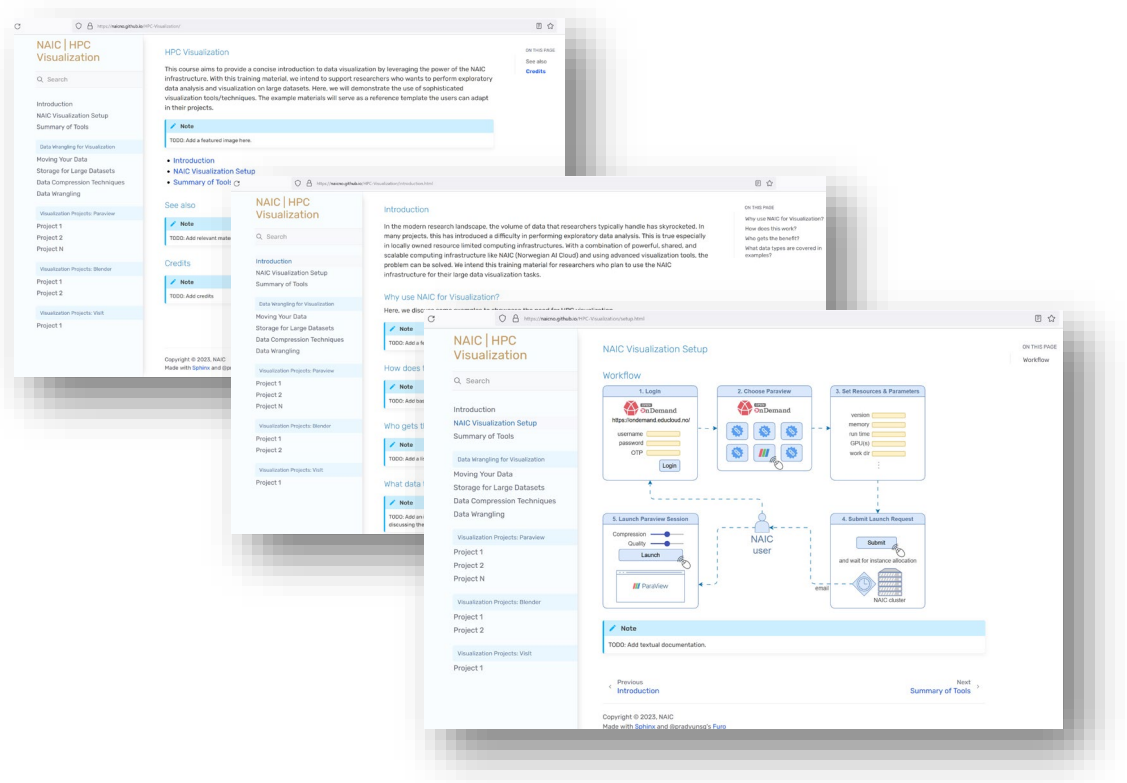


sherin.sugathan@dscience.uio.no

User Support

Documentation on use of ParaView under NAIC

https://naicno.github.io/HPC-Visualization/



sherin.sugathan@dscience.uio.no

User Support

NAIC User Support Portal

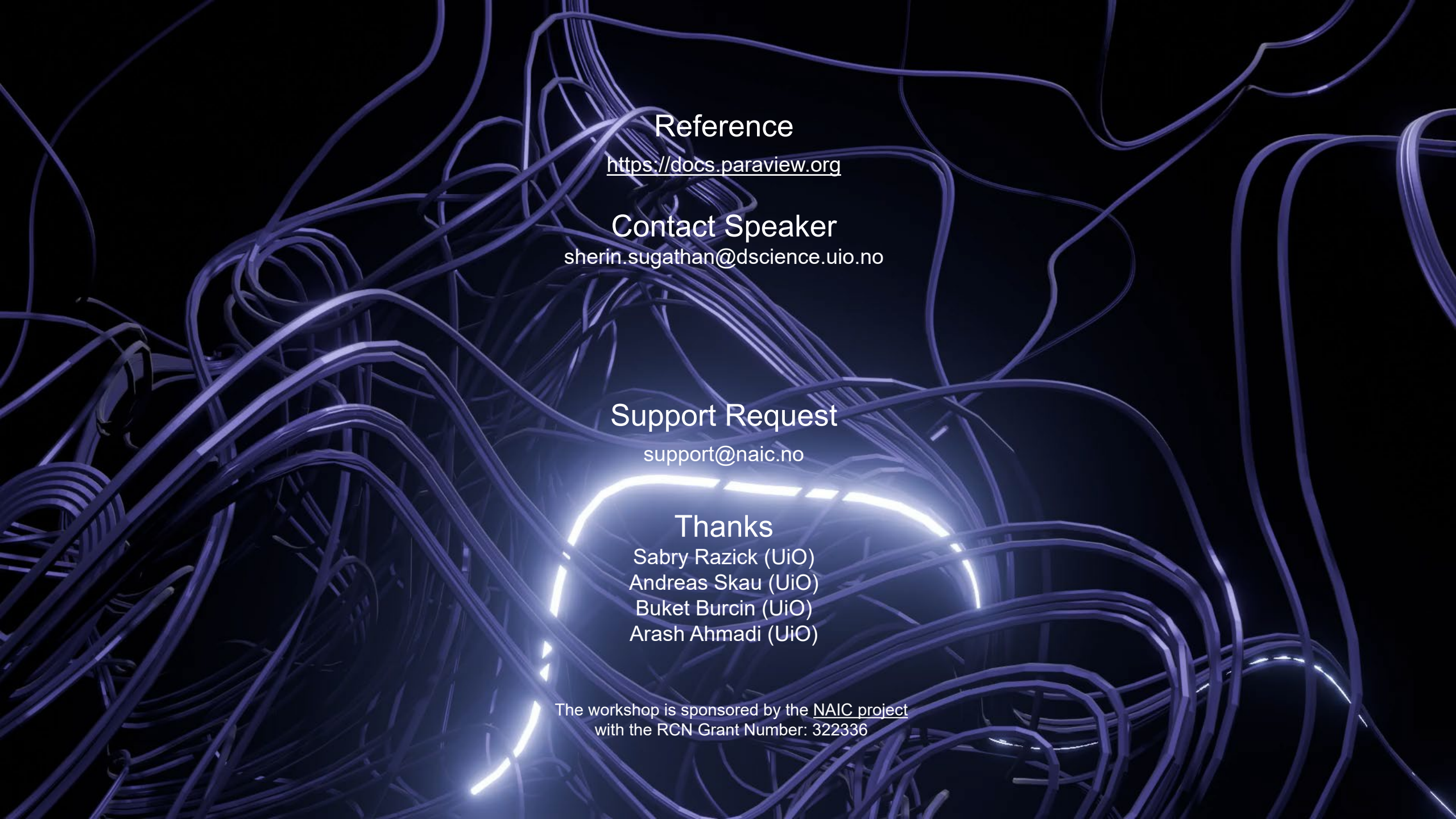
Norwegian AI Cloud
Support Desk

How to get in touch with the help center?

For administrative inquiries: contact@naic.no

For technical assistance: support@naic.no

For advanced user support: support@naic.no for detailed project guidance



Reference

https://docs.paraview.org

Contact Speaker

sherin.sugathan@dscience.uio.no

Support Request

support@naic.no

Thanks

Sabry Razick (UiO)
Andreas Skau (UiO)
Buket Burcin (UiO)
Arash Ahmadi (UiO)

The workshop is sponsored by the NAIC project
with the RCN Grant Number: 322336