

01 / 31 / 2024



# ALCF Developer Sessions

## Introducing Visualization on Polaris

Joseph Insley

Silvio Rizzi

Victor Mateevitsi

Janet Knowles

Argonne Leadership Computing Facility



## Here's what we plan to do...

- Provide high level overview of running visualization tools on Polaris
- Launching them for interactivity
- Running in batch
- Provide pointers to additional resources
- Time for hands-on

## Here's what we *Don't* plan to do...

- Give an overview of different types of data representations, or when to use which tools
- Give an in depth tutorial on using specific visualization tools
- Go into details about compiling specific tools

# Here's what we plan to do...

- VisIt
  - Overview
  - Launching on Polaris
- ParaView
  - Overview
  - Launching on Polaris
  - Blood Flow Example
  - Scripting
  - Running in Batch
- ImageMagick
- ffmpeg
- Hands-on

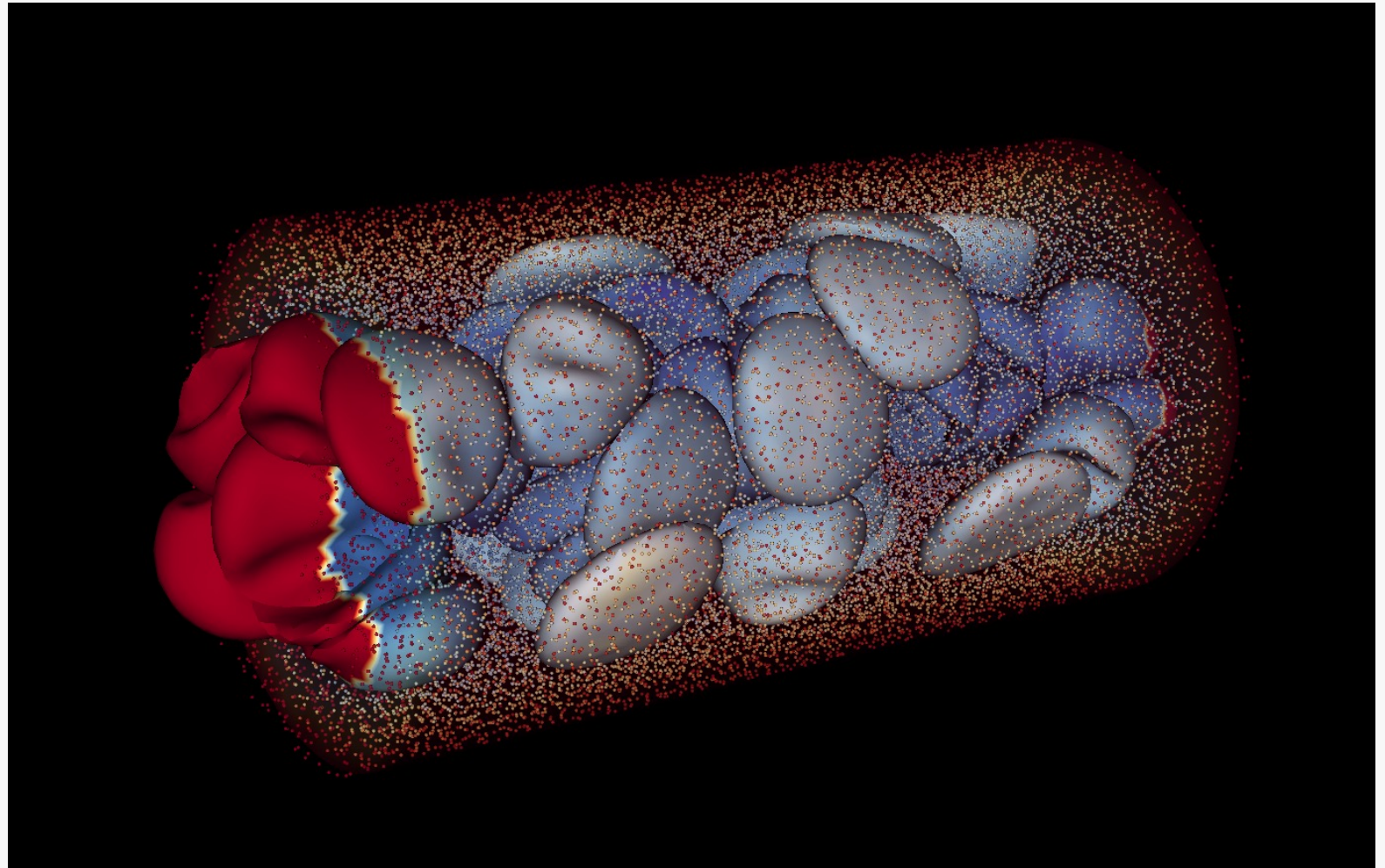
# Useful information not covered in this presentation

## ATPESC 2022 recorded sessions

- **ATPESC 2022 4.1 Data Analysis and Visualization Introduction:**  
<https://www.youtube.com/watch?v=2RaSDKtOq4w>
- **ATPESC 2022 4.3 Large Scale Visualization with ParaView:**  
<https://www.youtube.com/watch?v=MtnwvWkp6jE>
- **ATPESC 2022 4.4 Visualization and Analysis of HPC Simulation Data with VisIt:**  
<https://www.youtube.com/watch?v=rz1wMitvieY>
- **ATPESC 2022 4.5 Visualization and Analysis of HPC Simulation Data w/VisIt Con't.:** <https://www.youtube.com/watch?v=VI0eGrLzRmc>

# VisIt

Open Source  
Interactive  
Scalable  
Visualization  
Animation  
Analysis



# Visit on Polaris

- Latest version 3.3.3 (3.4 coming soon)
- Visit GUI connection to Polaris only
- Batch jobs not currently supported on Polaris

# Visit on Polaris

- Latest version 3.3.3 (3.4 coming soon)
- Visit GUI connection to Polaris only
- Batch jobs not currently supported on Polaris

# Visit on Polaris

- Latest version 3.3.3 (3.4 coming soon)
- **Visit GUI connection to Polaris only**
- Batch jobs not currently supported on Polaris



# Visit on Polaris

- Latest version 3.3.3 (3.4 coming soon)
- Visit GUI connection to Polaris only
- Batch jobs not currently supported on Polaris

# Visit on Polaris

- Latest version 3.3.3 (3.4 coming soon)
- Visit GUI connection to Polaris only
- Batch jobs not currently supported on Polaris

Information can be found at:

<https://docs.alcf.anl.gov/polaris/visualization/visit/>

# Visit: Connecting to Polaris

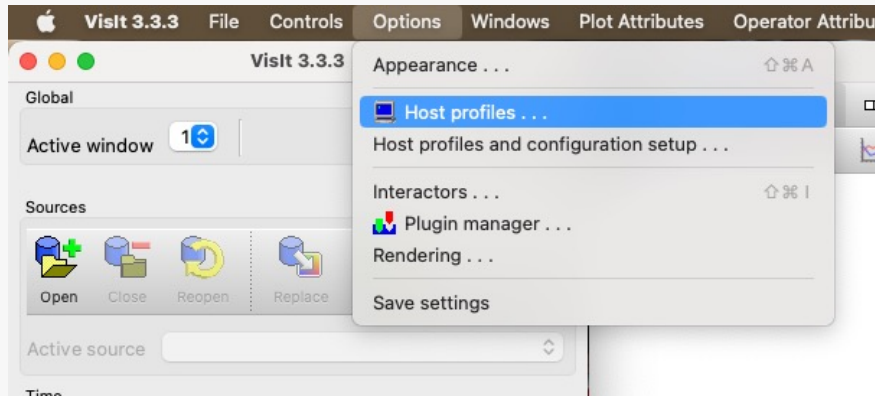
Download Polaris host profile:

[https://docs.alcf.anl.gov/polaris/visualization/scripts/host\\_anl\\_polaris.xml](https://docs.alcf.anl.gov/polaris/visualization/scripts/host_anl_polaris.xml)

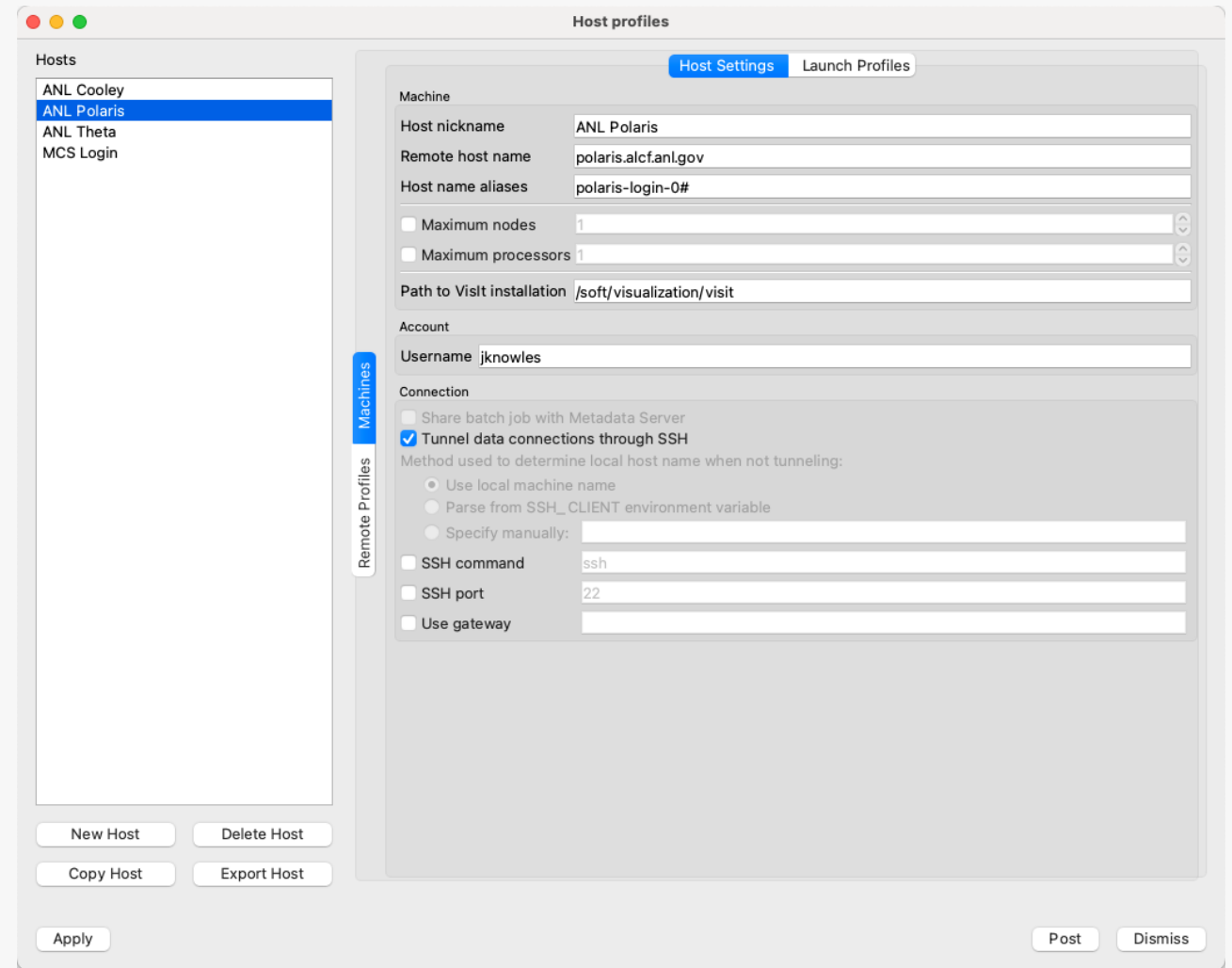
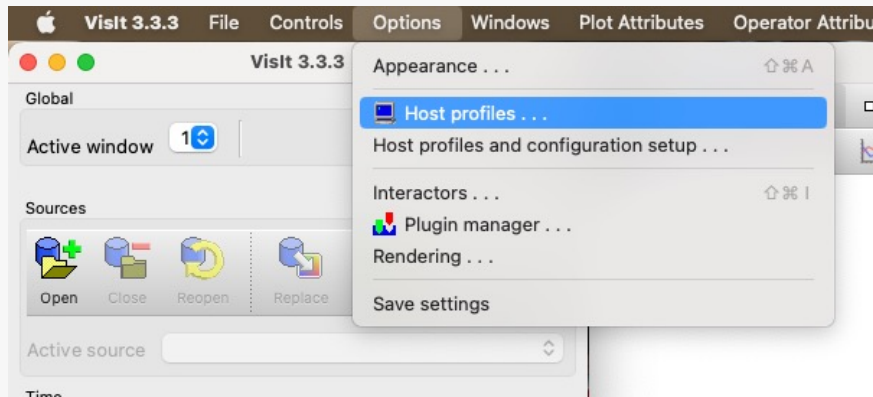
Copy this file to `~/.visit/hosts/host_anl_polaris.xml` (Mac/Linux)

```
▼<Object name="MachineProfile">
  <Field name="hostNickname" type="string">"ANL Polaris"</Field>
  <Field name="host" type="string">polaris.alcf.anl.gov</Field>
  <Field name="userName" type="string">notset</Field>
  <Field name="hostAliases" type="string">polaris-login-0#</Field>
  <Field name="directory" type="string">/soft/visualization/visit</Field>
  <Field name="shareOneBatchJob" type="bool">>false</Field>
  <Field name="sshPortSpecified" type="bool">>false</Field>
  <Field name="sshPort" type="int">22</Field>
  <Field name="sshCommandSpecified" type="bool">>false</Field>
  <Field name="sshCommand" type="stringVector">"ssh" </Field>
  <Field name="useGateway" type="bool">>false</Field>
  <Field name="gatewayHost" type="string"/>
  <Field name="clientHostDetermination" type="string">MachineName</Field>
  <Field name="manualClientHostName" type="string"/>
  <Field name="tunnelSSH" type="bool">>true</Field>
  <Field name="maximumNodesValid" type="bool">>false</Field>
```

# Connecting to Polaris: Set up the Host Profile



# Connecting to Polaris: Set up the Host Profile



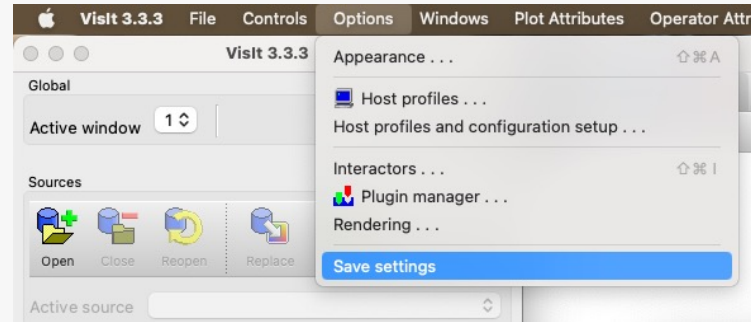
# Connecting to Polaris: Set up the Host Profile

Number of Processors  
Number of Nodes  
Bank / Account  
Time Limit

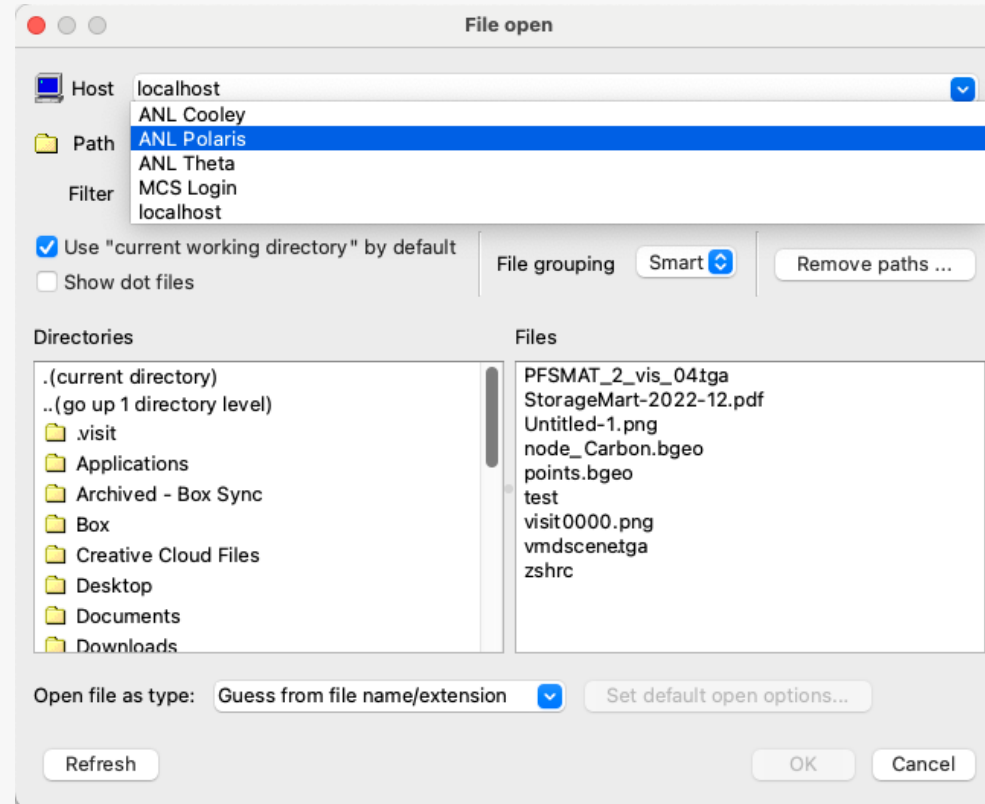
The screenshot shows a web interface for configuring host profiles. On the left, a list of hosts includes ANL Cooley, ANL Polaris, ANL Theta, and MCS Login. The main area is titled 'Host profiles' and has tabs for 'Host Settings' and 'Launch Profiles'. The 'Launch Profiles' tab is active, showing a list of profiles: 'parallel' (checked) and 'serial' (unchecked). Below this list are buttons for 'New Profile', 'Delete Profile', 'Copy Profile', and 'Make Default'. A vertical sidebar on the left of the main area is labeled 'Remote Profiles' and has a 'Machines' tab. The 'Machines' tab is active, showing a 'Settings' section with sub-tabs for 'Parallel' and 'GPU Acceleration'. The 'Parallel' sub-tab is active, showing a checked box for 'Launch parallel engine' and a 'Launch' button. Below this are fields for 'Parallel launch method' (qsub/aprun), 'Partition / Pool / Queue' (debug), and a 'Defaults' section with fields for 'Number of processors' (4), 'Number of nodes' (1), 'Bank / Account' (visualization), 'Time Limit' (1:00:00), and 'Machine File' (\$PBS\_NODEFILE). To the right of the 'Defaults' section is a 'Constraints' section with 'Add row' and 'Delete row' buttons and a table with columns for 'Nodes' and 'Processors'. At the bottom of the interface are buttons for 'New Host', 'Delete Host', 'Copy Host', 'Export Host', 'Apply', 'Post', and 'Dismiss'.

# Connecting to Polaris: Set up the Host Profile

Save your settings

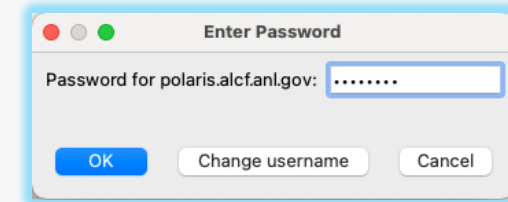
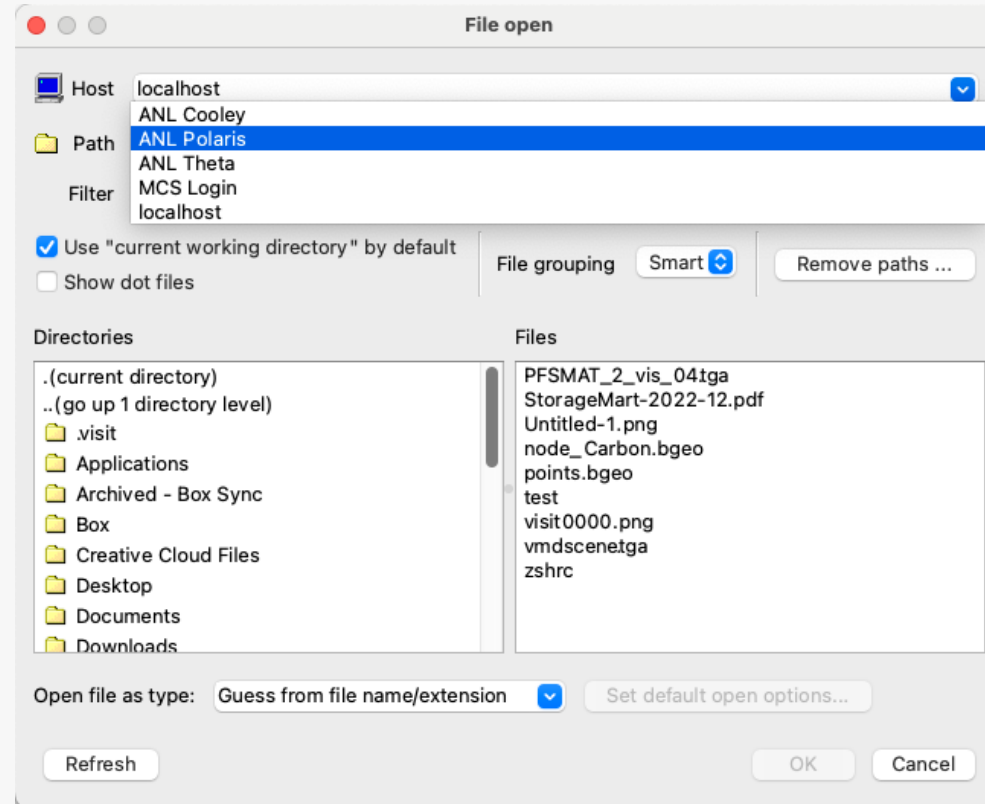


# Connecting to Polaris

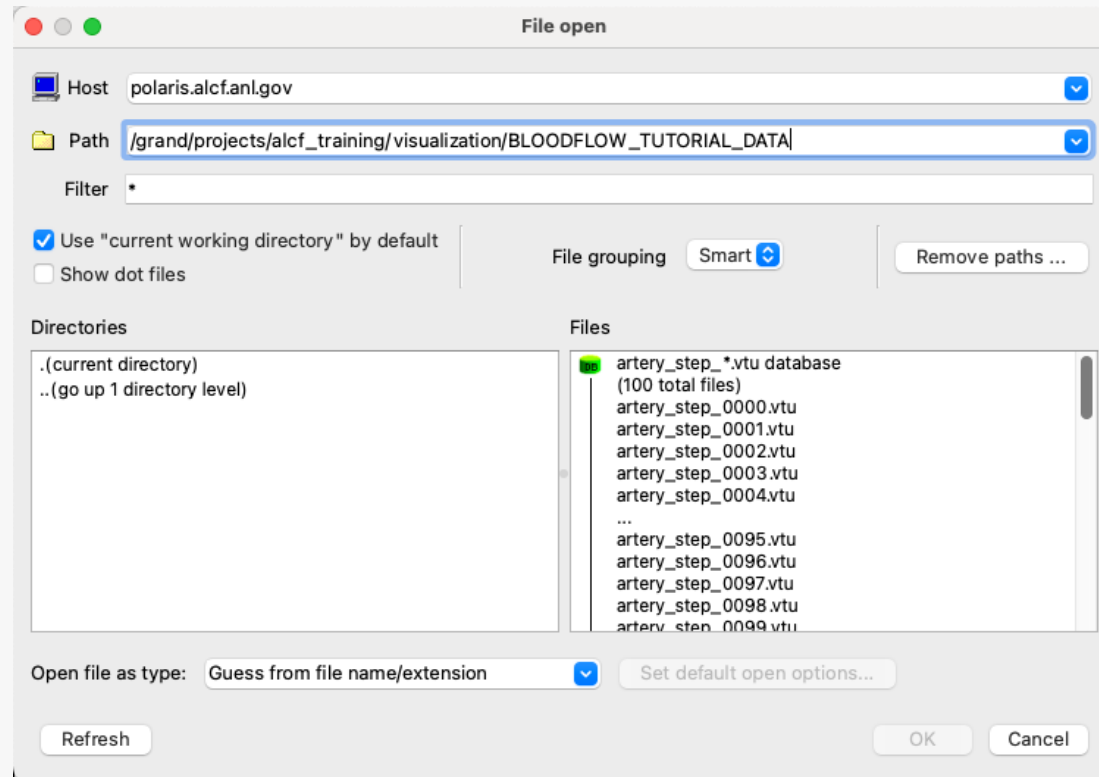




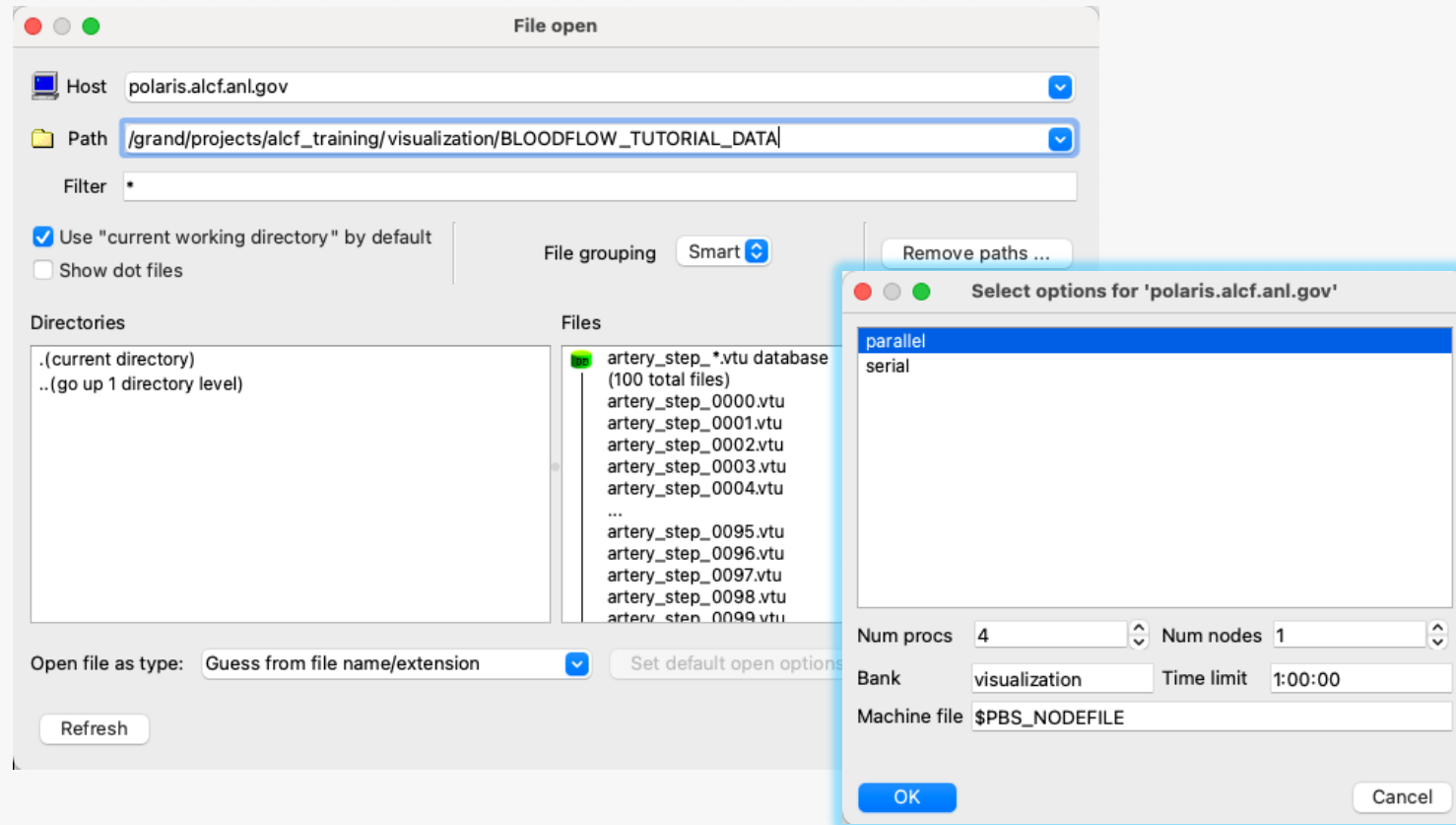
# Connecting to Polaris



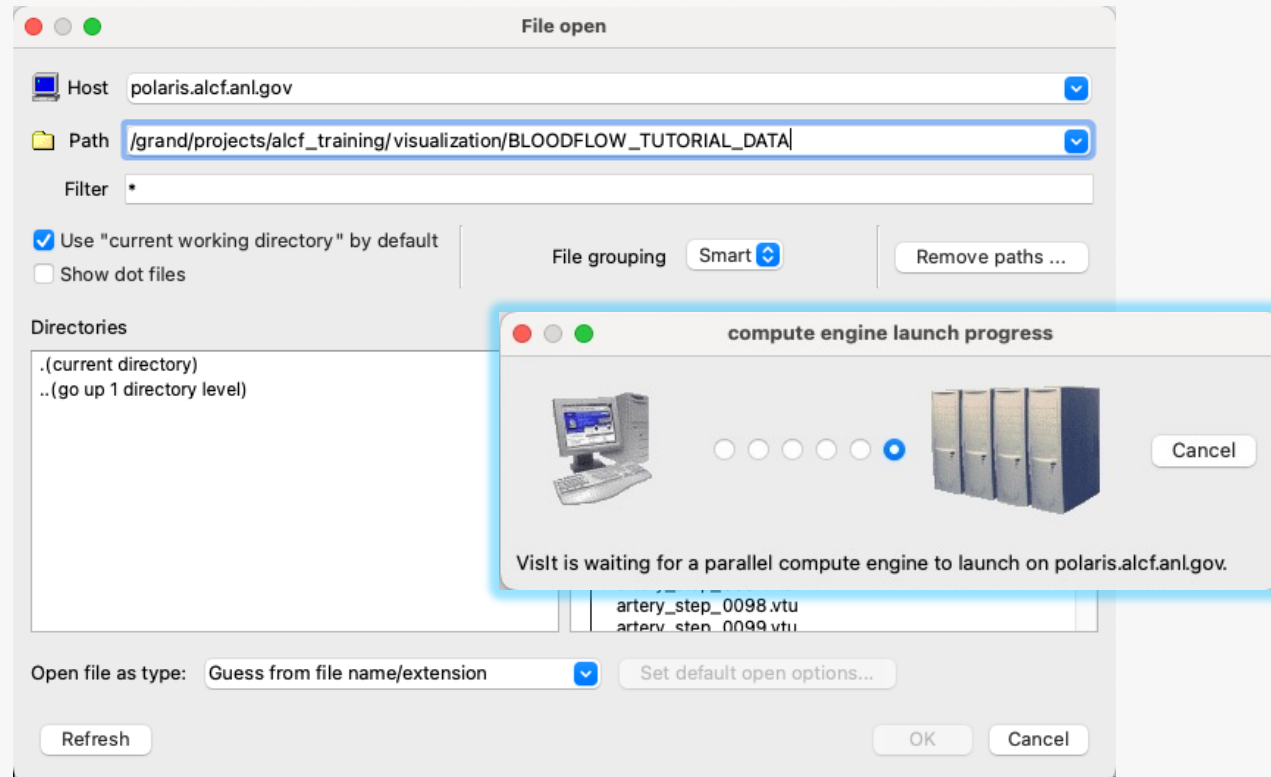
# Connecting to Polaris



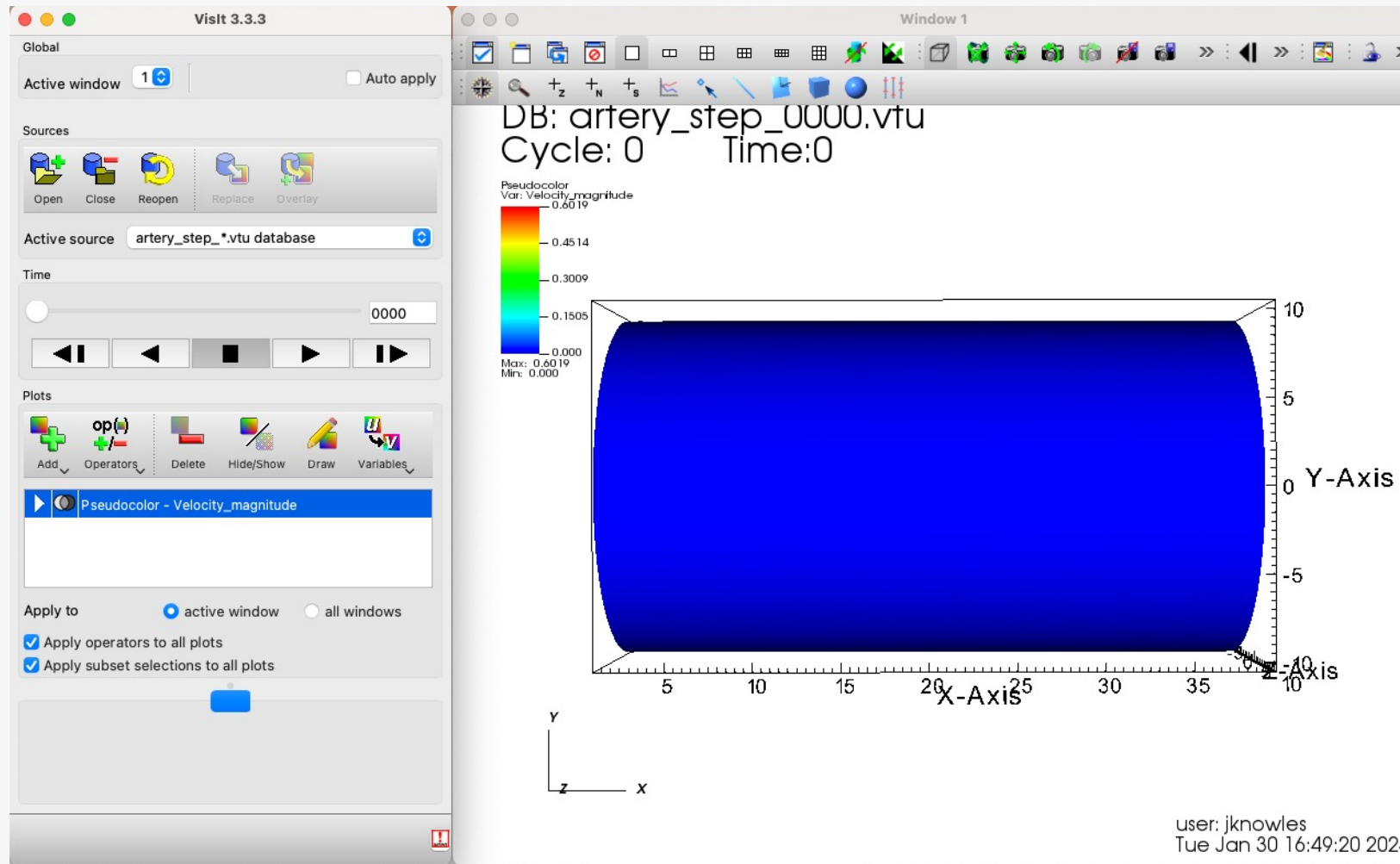
# Connecting to Polaris



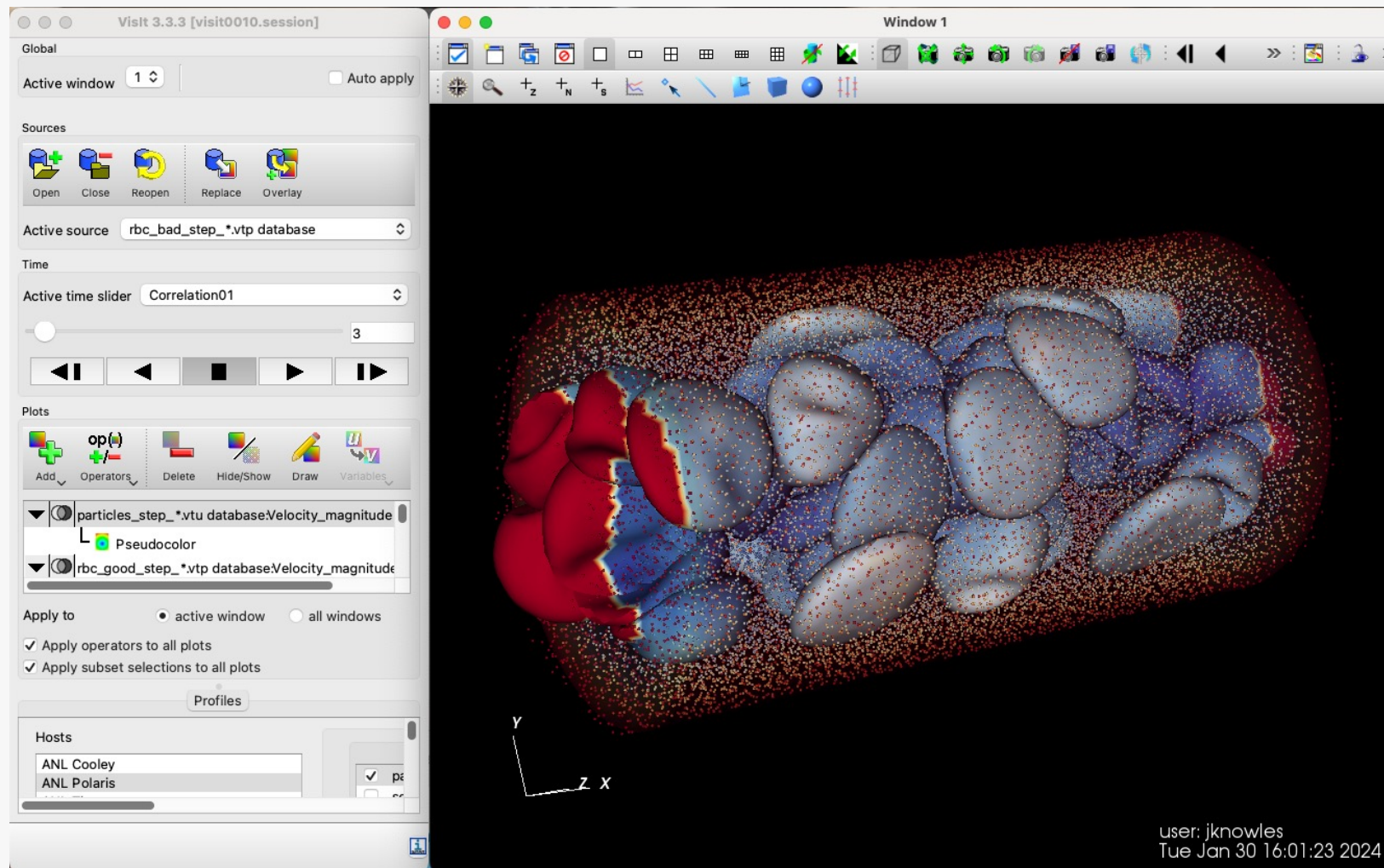
# Connecting to Polaris



# Connecting to Polaris



# Connecting to Polaris



# ParaView on Polaris

- Download client from <https://www.paraview.org/download/>
- ALCF user guides here <https://docs.alcf.anl.gov/polaris/visualization/paraview/>
- How to check ParaView versions available on Polaris.
- From an ssh session run:

```
module avail paraview
```

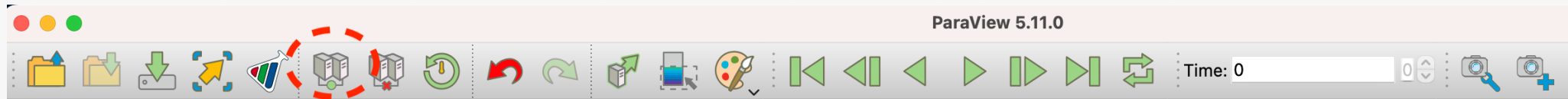
```
----- /soft/modulefiles -----  
paraview/paraview-5.11.1-mesa  paraview/paraview-5.11.2-mesa  paraview/paraview-5.12.0-RC1-mesa  
(D)
```

- The client version you download from Kitware must match one of the installed versions, 5.11.1, 5.11.2, or 5.12.0-RC1 at this time. **Ignore the mesa suffix**

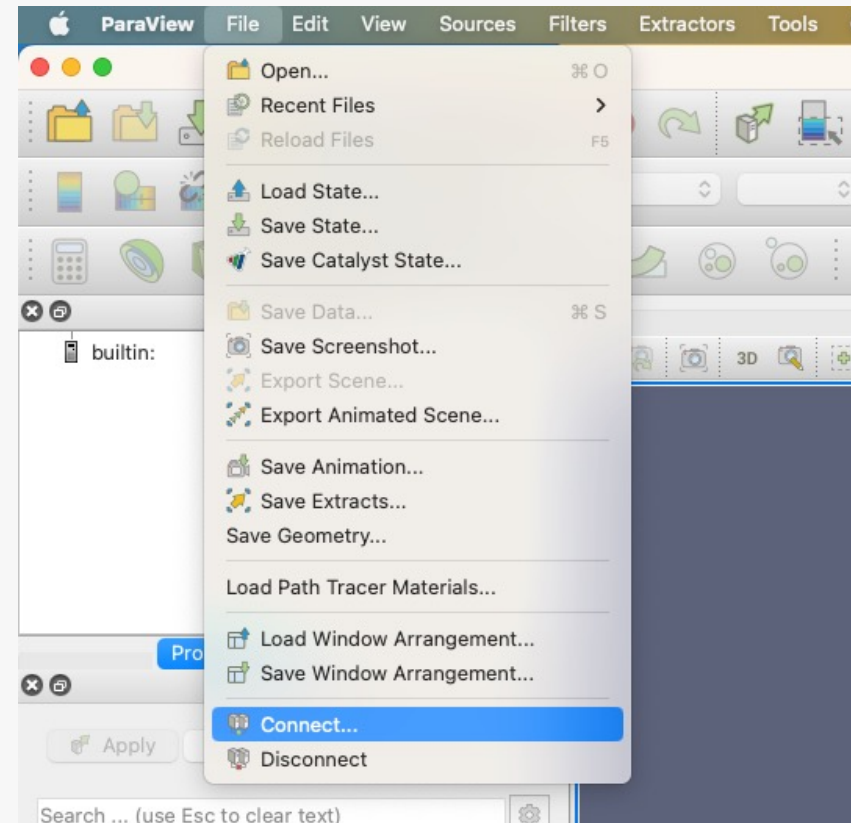
# Server Configuration

## 1. Select Connect

From the ParaView client choose to connect to a server by either clicking on the "Connect" icon in the menu bar



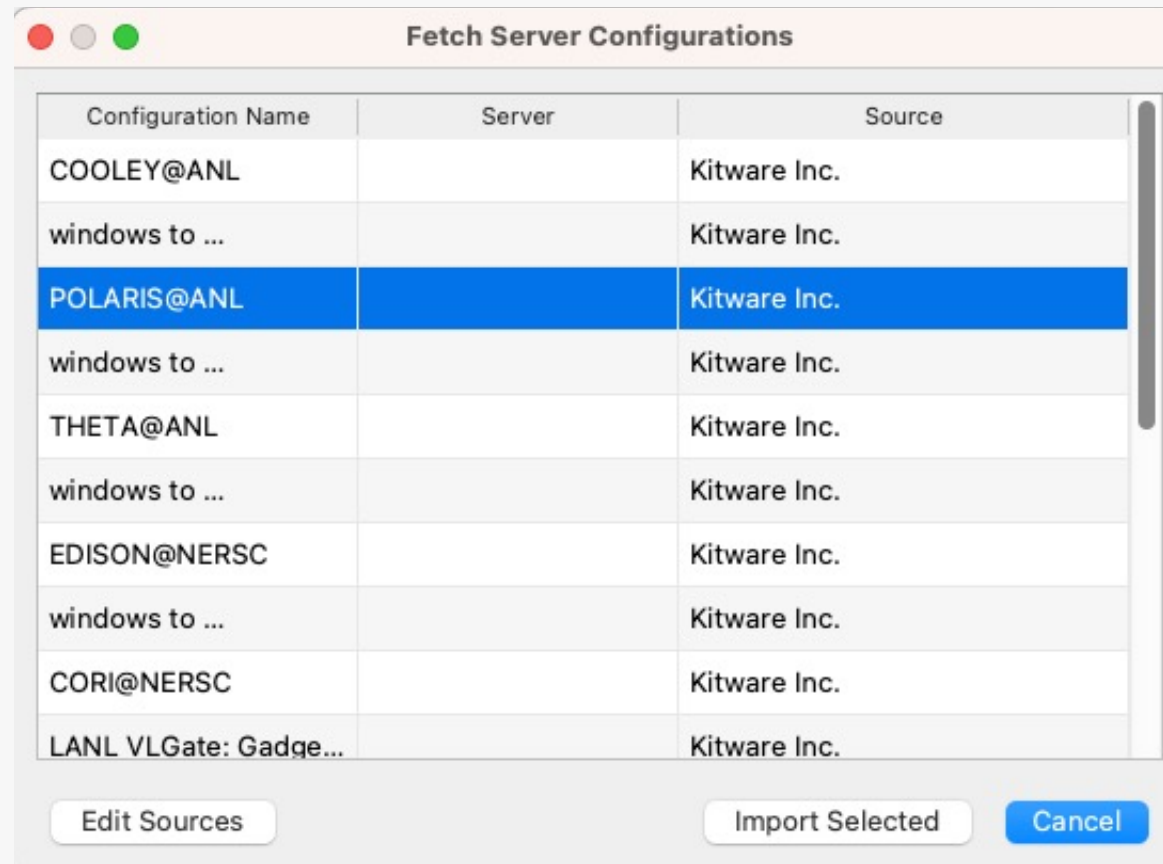
or selecting File->Connect from the main menu





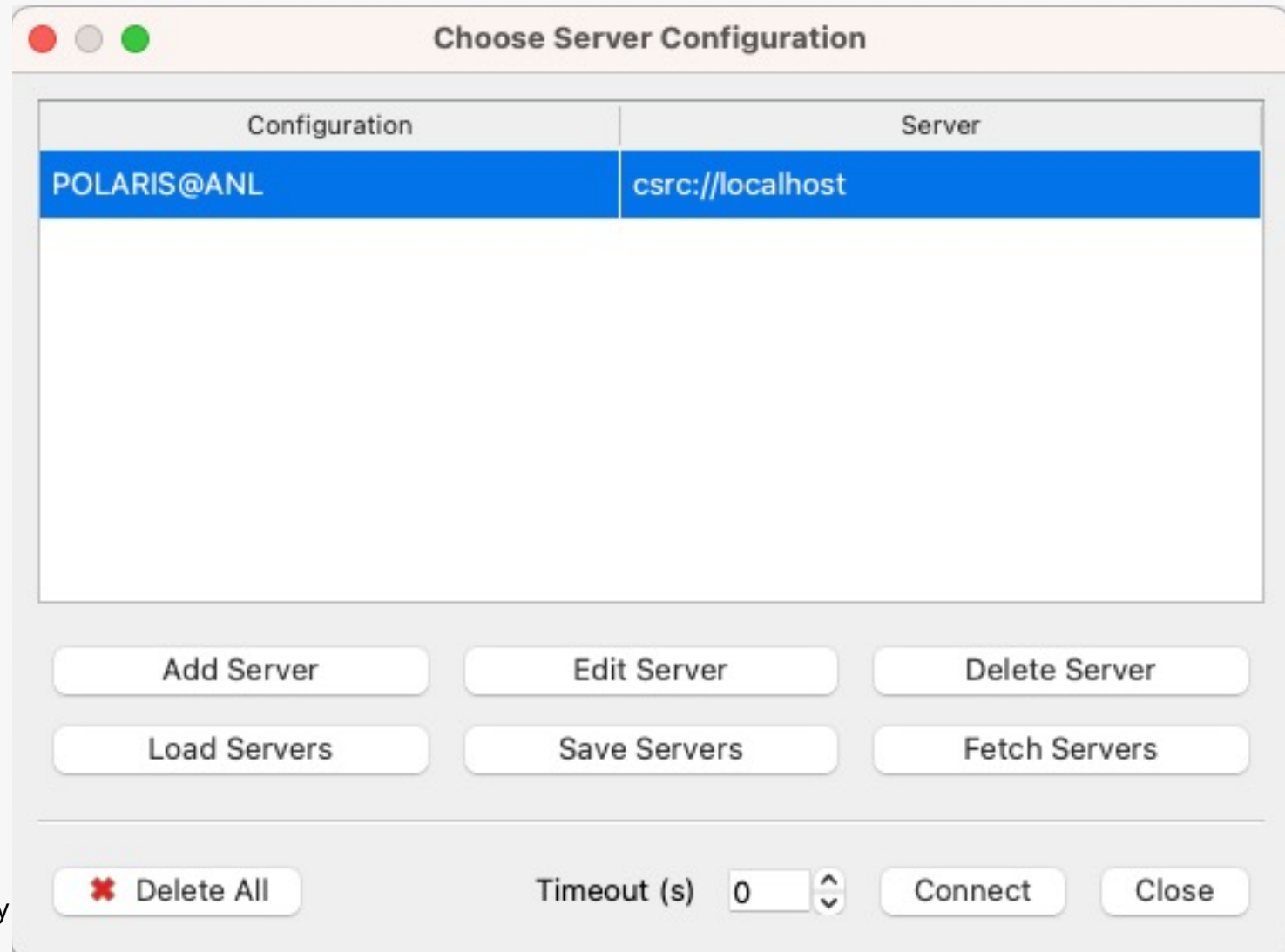
## 2. Set Up Servers (first time only)

In the File->Connect menu press the button named "Fetch Servers" and select POLARIS@ANL. Windows users should select "windows to POLARIS@ANL". Press "Import Selected"



### 3. Use ParaView

You can now select POLARIS@ANL in the File->Connect menu and press Connect



### 3. Use ParaView (cont'd)

ALCF username

5.11.2-mesa

alcf\_training

R1222649

Connection Options for "POLARIS@ANL"

Xterm executable  ...

SSH executable  ...

Remote machine

Username

ParaView version

Client port  ^ v

Server port  ^ v

Number of nodes to reserve  ^ v

Number of ranks per node  ^ v

Number of minutes to reserve  ^ v

Account

Queue

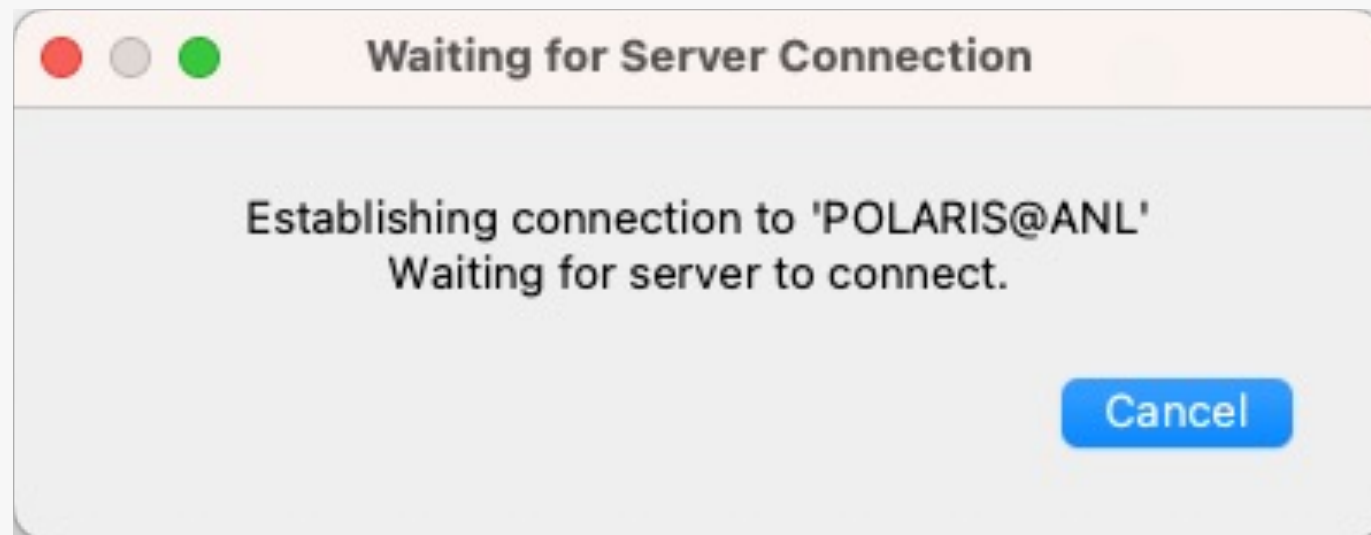
File Systems

Job name

Cancel OK

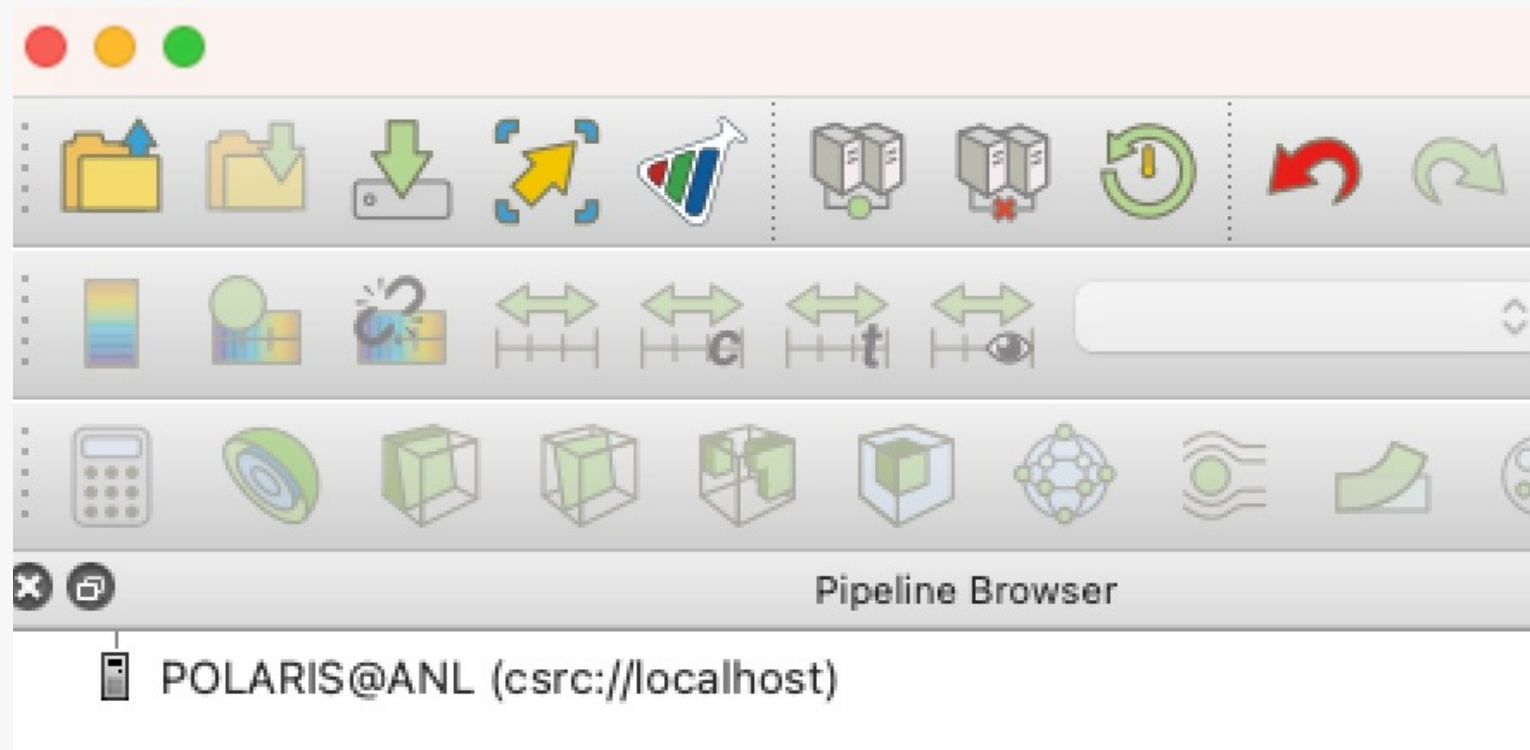
### 3. Use ParaView (cont'd)

An ssh connection will be established with a Polaris login node and a password will be requested in a terminal. After you enter your password, a job will be queued and you will see a window like this:



### 3. Use ParaView (cont'd)

When the job is launched on the compute nodes, the previous window will go away and ParaView will show it is connected to Polaris in its Pipeline Browser:



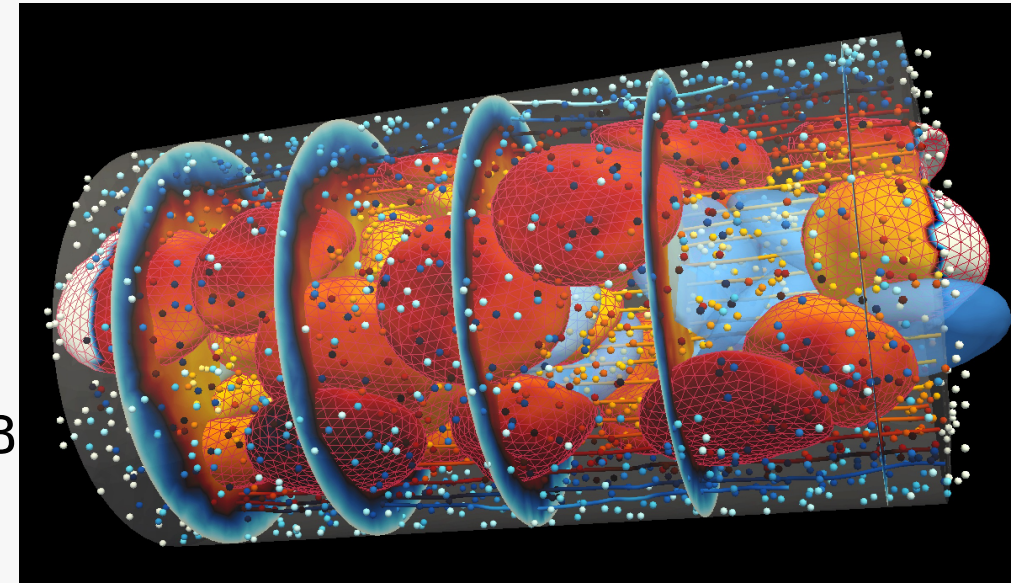
# Red Blood Cell Data

## Tutorial:

– <https://docs.alcf.anl.gov/polaris/visualization/paraview-tutorial/>

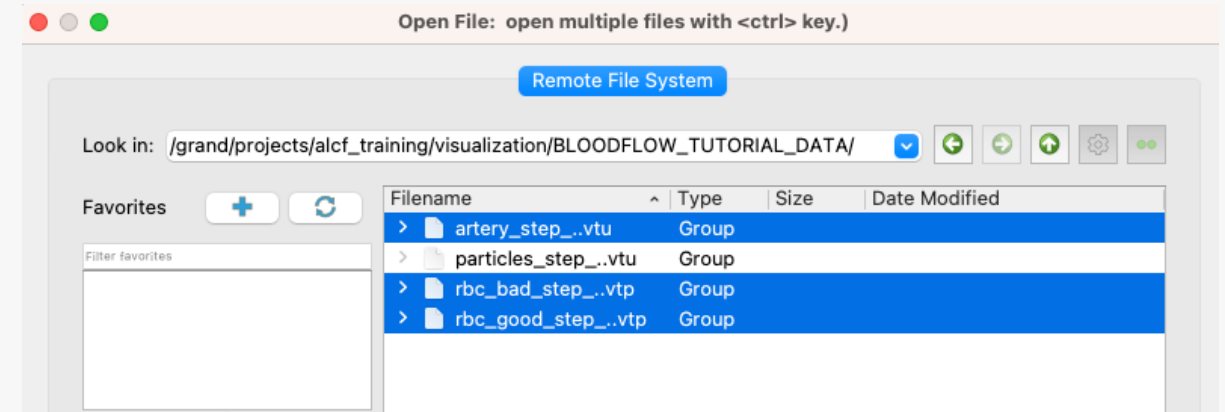
## Load Data:

```
/grand/projects/alcf_training/visualization/BLOODFLOW_TUTORIAL_DATA/
```



## Files:

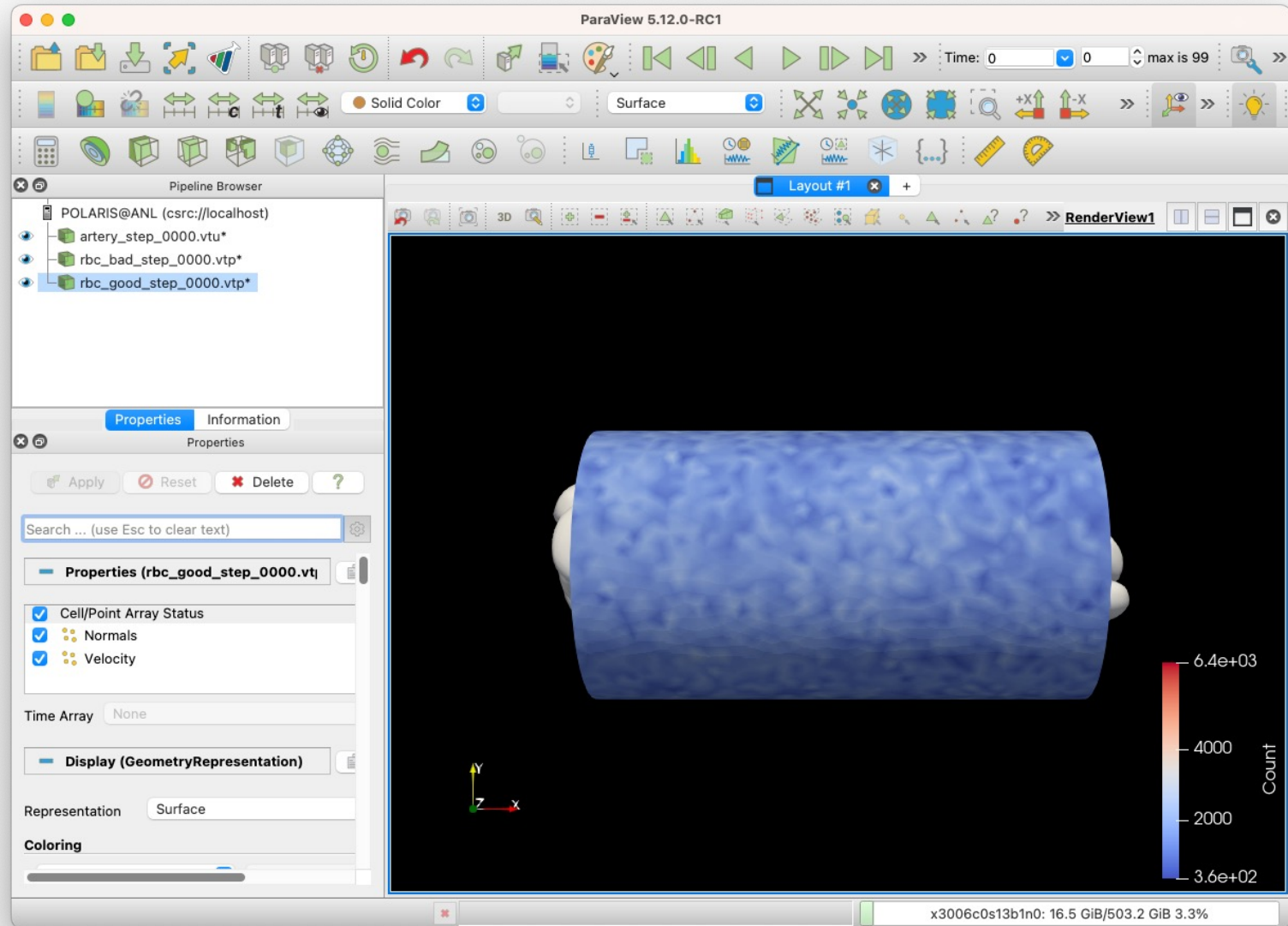
```
> artery_step_...vtu  
> rbc_bad_step_..vtu  
> rbc_good_step_...vtu
```



# Red Blood Cell Data

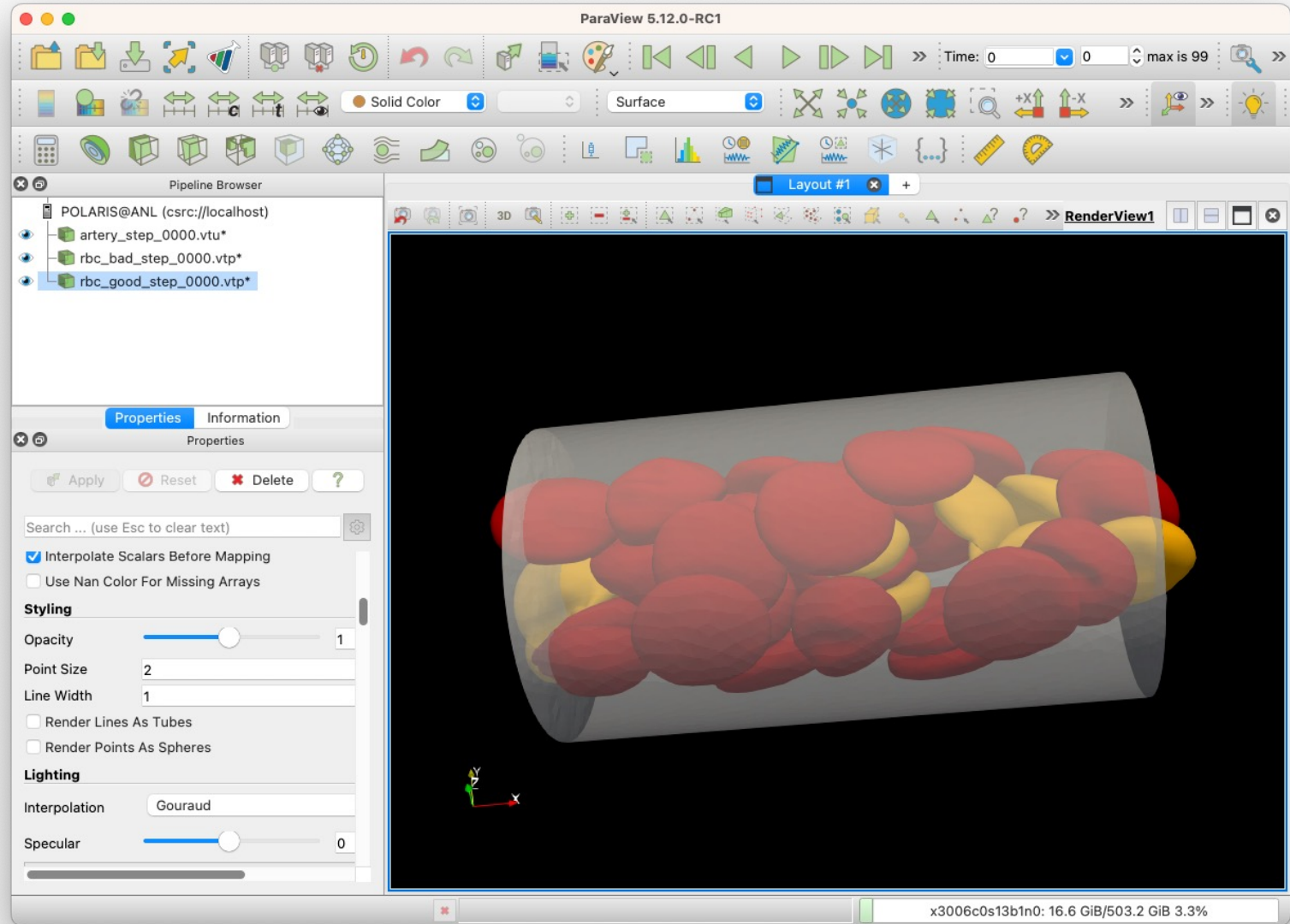
## Files:

- > artery\_step\_...vtu
- > rbc\_bad\_step\_...vtu
- > rbc\_good\_step\_...vtu



# Red Blood Cell Data

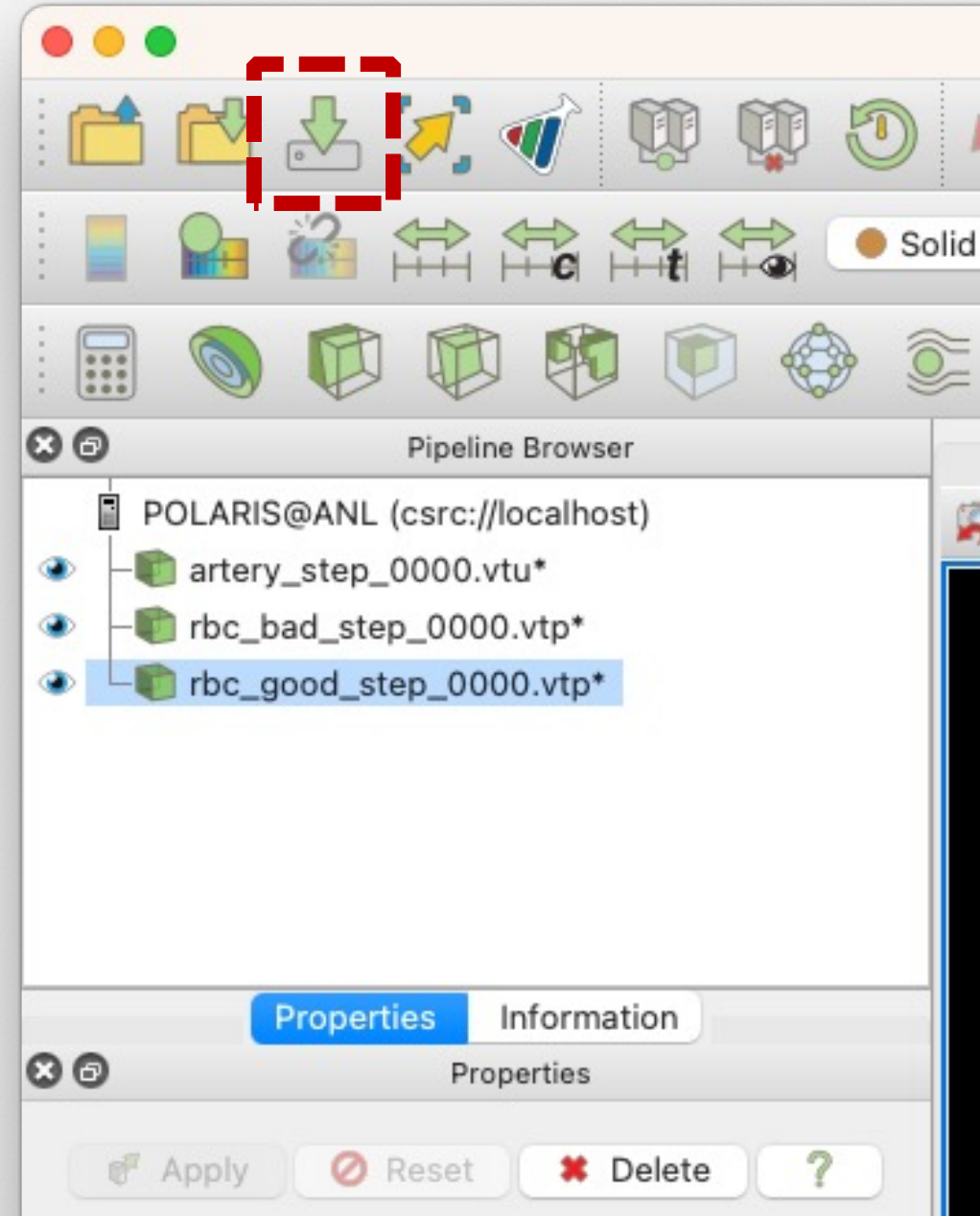
- > artery\_step\_...vtu
  - Color: White
  - Opacity: 0.3
- > rbc\_bad\_step\_..vtu
  - Color: Gold
- > rbc\_good\_step\_...vtu
  - Color: Red





# ParaView States and Scripting

- Choose: File → Save State...
  - .pvsm (for restoring state in interactive mode)
  - saved on the client side
- Choose: File → Save State...
  - .py (for use with pvbatch)
  - saved on the client side
- Edit .py script
  - short example, loop over time steps, saving images
  - `/grand/projects/alcf_training/visualization/SCRIPTS/blood_tutorial_01_2024.py`



# ParaView States and Scripting

```
import sys
start_frame=int(sys.argv[1])
num_frames=int(sys.argv[2])
total_steps = 100

IMAGE_DIR = "/grand/projects/alcf_training/visualization/FRAMES/TEST_01"
DATA_DIR = "/grand/projects/alcf_training/visualization/BLOODFLOW_TUTORIAL_DATA"

ARTERY_FILES = []
RBC_BAD_FILES = []
RBC_GOOD_FILES = []

for i in range(0, total_steps):
    temp_name = "%s/artery_step_%04d.vtu" % (DATA_DIR, i)
    ARTERY_FILES.append(temp_name)
    temp_name = "%s/rbc_bad_step_%04d.vtp" % (DATA_DIR, i)
    RBC_BAD_FILES.append(temp_name)
    temp_name = "%s/rbc_good_step_%04d.vtp" % (DATA_DIR, i)
    RBC_GOOD_FILES.append(temp_name)
```



# ParaView States and Scripting

```
# create a new 'XML Unstructured Grid Reader'  
artery_step_0000vtu =  
    XMLUnstructuredGridReader(registrationName='artery_step_0000.vtu*',  
    FileName=ARTERY_FILES[start_frame])  
artery_step_0000vtu.PointArrayStatus = ['Count', 'Velocity']  
artery_step_0000vtu.TimeArray = 'None'  
  
# create a new 'XML PolyData Reader'  
rbc_bad_step_0000vtp =  
    XMLPolyDataReader(registrationName='rbc_bad_step_0000.vtp*',  
    FileName=RBC_BAD_FILES[start_frame])  
rbc_bad_step_0000vtp.PointArrayStatus = ['Velocity', 'Normals']  
rbc_bad_step_0000vtp.TimeArray = 'None'  
  
# create a new 'XML PolyData Reader'  
rbc_good_step_0000vtp =  
    XMLPolyDataReader(registrationName='rbc_good_step_0000.vtp*',  
    FileName=RBC_GOOD_FILES[start_frame])  
rbc_good_step_0000vtp.PointArrayStatus = ['Velocity', 'Normals']  
rbc_good_step_0000vtp.TimeArray = 'None'
```

# ParaView States and Scripting

```
XRES = 1920
```

```
YRES = 1080
```

```
renderView1.ViewSize = [XRES, YRES]
```

```
renderView1.Background = [0.0, 0.0, 0.0]
```

```
LoadPalette(paletteName='BlackBackground')
```

```
for i in range(start_frame, start_frame+num_frames):
```

```
artery_step_0000vtu.FileName=ARTERY_FILES[i]
```

```
rbc_bad_step_0000vtp.FileName=RBC_BAD_FILES[i]
```

```
rbc_good_step_0000vtp.FileName=RBC_GOOD_FILES[i]
```

```
IMAGE_FILE_NAME="%s/frame_%04d.png" % (IMAGE_DIR,i)
```

```
print("Save: ", IMAGE_FILE_NAME)
```

```
SaveScreenshot(IMAGE_FILE_NAME, renderView1, ImageResolution=[XRES, YRES])
```

# Running ParaView Script from Interactive Job

Submit an interactive job

```
> qsub -I -A alcf_training -q R1222649 -l select=1 -l walltime=1:00:00 -l filesystems=home:grand
```

Load the ParaView module

```
x3006c0s19b0n0:~> module load paraview
```

Run the script using pvbatch

```
x3006c0s19b0n0:~> pvbatch /<path>/blood_tutorial_01_2024.py 0 100
```

# Running ParaView Script from Batch Job

Create a shell script to run our job

```
/grand/projects/alcf_training/visualization/SCRIPTS/run_pv_script_01.sh
```

```
#!/bin/bash -l  
start_frame=$1  
num_frames=$2
```

```
module load paraview  
pvbatch
```

```
    /grand/projects/alcf_training/visualization/SCRIPTS/blood_tutorial_01_2024.py  
    $start_frame $num_frames
```

# Running ParaView Script from Batch Job

Submit the job to the scheduler

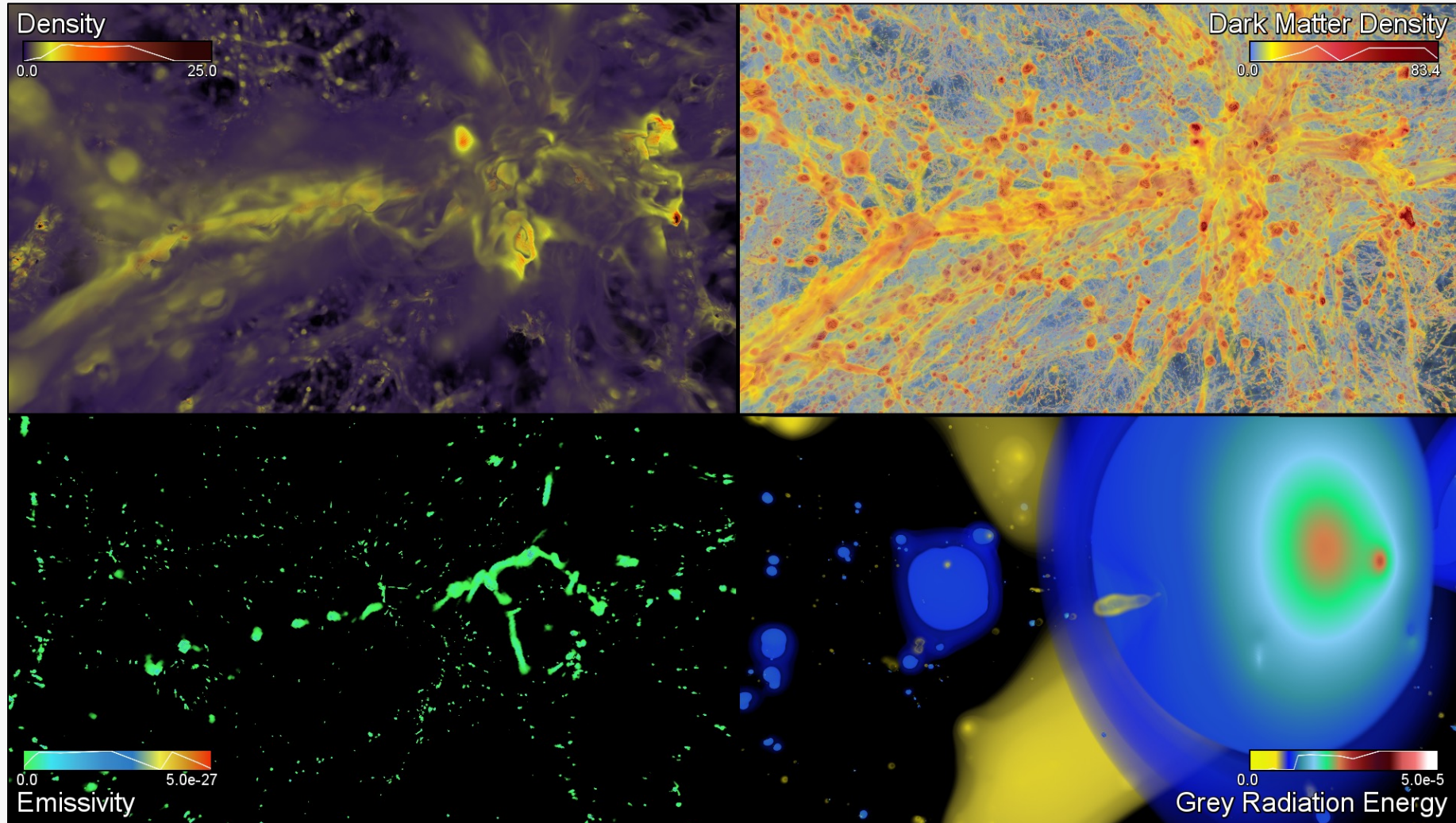
```
> qsub -A alcf_training -q R1222649 -l select=1 -l walltime=1:00:00 -l  
filesystems=home:grand --  
/grand/projects/alcf_training/visualization/SCRIPTS/run_pv_script_01.sh 0 50  
  
> qsub -A alcf_training -q R1222649 -l select=1 -l walltime=1:00:00 -l  
filesystems=home:grand --  
/grand/projects/alcf_training/visualization/SCRIPTS/run_pv_script_01.sh 50 50
```



# Annotation, compositing, scaling...

## ImageMagick

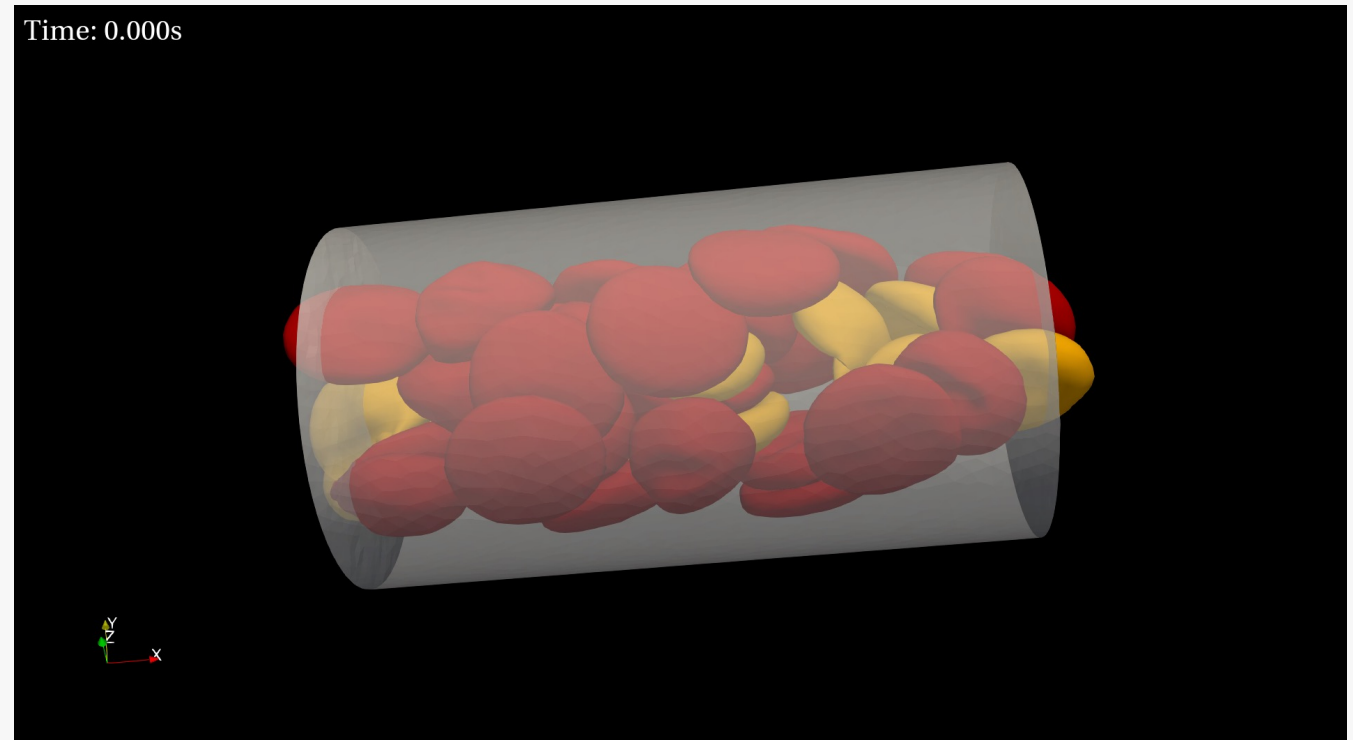
- convert, composite, annotate, montage, etc.



# Annotation, compositing, scaling...

## ImageMagick

```
> module load imagemagick  
> magick in_image.png -fill white -gravity northwest -pointsize 40 -annotate  
+15+15 'Time: 0.000s' -depth 8 out_image.png
```



# Annotation, compositing, scaling...

## ImageMagick

```
/grand/projects/alcf_training/visualization/SCRIPTS/annotate_frames_01.py
```

```
> module load imagemagick
```

```
> which magick
```

```
import sys, os
```

```
def main():
```

```
    start_frame=int(sys.argv[1])
```

```
    num_frames=int(sys.argv[2])
```

```
    magick_exec="/path/to/magick"
```

```
    in_dir = "/path/to/in_frames"
```

```
    out_dir="/path/to/out_frames"
```

```
    font_size = 40
```

```
    time_start = 0.0
```

```
    time_delta = 0.225
```

# Annotation, compositing, scaling...

## ImageMagick

```
/grand/projects/alcf_training/visualization/SCRIPTS/annotate_frames_01.py
```

```
for i in range(start_frame, start_frame+num_frames):
    time_val = "Time: %0.03fs" % (time_start + (time_delta*i))
    in_frame = "%s/frame_%04d.png" % (in_dir, i)
    out_frame = "%s/frame_%04d.png" % (out_dir, i)
    magick_cmd = "%s %s -fill white -gravity northwest -pointsize %d -
annotate +15+15 '%s' -depth 8 %s" % (magick_exec, in_frame, font_size, time_val,
out_frame )
    print ("magick_cmd= ", magick_cmd)
    os.system(magick_cmd)
```

# Movie Encoding

ffmpeg

```
> module load ffmpeg
```

```
> ffmpeg -r 25 -start_number 0 -i /path/frame_%04d.png  
-r 25 -pix_fmt yuv420p movie.mp4
```

# Reservation information

Allocation: alcf\_training

Queue: R1222649

# HANDS-ON & QUESTIONS?

Joe Insley  
insley@anl.gov

Silvio Rizzi  
srizzi@anl.gov

Janet Knowles  
jknowles@anl.gov

Victor Mateevitsi  
vmateevitsi@anl.gov

[www.anl.gov](http://www.anl.gov)