

December User Conference Call Analysis Analysis and Visualization Tools at OLCF

Benjamin Hernandez
Computer Scientist
Analytics and AI Methods at Scale Group
OLCF

Dec 8th, 2021

ORNL is managed by UT-Battelle LLC for the US Department of Energy



U.S. DEPARTMENT OF
ENERGY

Contents

- General overview of analysis infrastructure at OLCF
- Visualization Tools
 - Remote Desktops
 - TurboVNC and NiceDCV
- Scientific Visualization Tools Overview
 - Paraview
 - Visit

Tools Overview

- docs.olcf.ornl.gov

← → ↻ https://docs.olcf.ornl.gov ☆ 🔒 📧

OAK RIDGE | LEADERSHIP COMPUTING FACILITY
National Laboratory

🏠 OLCF User Documentation

Search docs

- New User Quick Start
- ⊕ Accounts and Projects
- ⊕ Connecting
- ⊕ Systems
- ⊕ Services and Applications
- ⊕ Data Storage and Transfers
- ➡ **Software**
 - Software News
 - ⊕ ML/DL & Data Analytics
 - ⊕ Python on OLCF Systems
 - ⊕ Profiling Tools
 - ⊕ User-Managed Software
 - ⊕ Workflows
 - ⊕ E4S Software Stack
 - ⊕ Spack Environments
 - ⊕ Visualization Tools
- ⊕ Training
- ⊕ Contributing to these docs

» OLCF User Documentation [Edit on GitHub](#) | [OLCF Home Page](#)

OLCF User Documentation

Note

The [OLCF User Assistance Center](#) remains open and available for support. Users should follow normal support procedures when reporting issues or requesting help.

- [Submit a Support Ticket](#)
- Email us at help@olcf.ornl.gov

This technical documentation is a reference for the user community to efficiently use OLCF compute and storage resources.

Have an idea to improve this documentation? See [Contributing to these docs](#).

Have a question? Write to us at – help@olcf.ornl.gov – and consider the guidelines for [Getting Help](#).

New User Quick Start

- [New User Quick Start](#)

Accounts

- [Accounts and Projects](#)

Tools Overview. Python at OLCF

The screenshot displays the OLCF Python documentation website. The browser address bar shows the URL <https://docs.olcf.ornl.gov/software/python/index.html>. The page header includes the OAK RIDGE National Laboratory logo and the text 'LEADERSHIP COMPUTING FACILITY'. Below the header is a search bar labeled 'Search docs'. The sidebar on the left contains a list of navigation links: 'New User Quick Start', 'Accounts and Projects', 'Connecting', 'Systems', 'Services and Applications', 'Data Storage and Transfers', 'Software', 'Software News', 'ML/DL & Data Analytics', 'Python on OLCF Systems' (which is expanded to show 'Overview', 'OLCF Python Guides', 'Module Usage', 'How to Run', and 'Best Practices'), 'Profiling Tools', 'User-Managed Software', 'Workflows', 'E4S Software Stack', 'Spack Environments', and 'Visualization Tools'. The main content area has a breadcrumb trail: '» Software » Python on OLCF Systems'. It also includes links to 'Edit on GitHub' and 'OLCF Home Page'. The main heading is 'Python on OLCF Systems' followed by a subheading 'Overview'. The text in the overview section states: 'In high-performance computing, [Python](#) is heavily used to analyze scientific data on the system. Some users require specific versions of Python or niche scientific packages to analyze their data, which may further depend on numerous other Python packages. Because of all the dependencies that some Python packages require, and all the types of data that exist, it can be quite troublesome to get different Python installations to “play nicely” with each other, especially on an HPC system where the system environment is complicated. [Conda](#), a package and virtual environment manager from the [Anaconda](#) distribution, helps alleviate these issues.' Below this, it says: 'Conda allows users to easily install different versions of binary software packages and any required libraries appropriate for their computing platform. The versatility of conda allows a user to essentially build their own isolated Python environment, without having to worry about clashing dependencies and other system installations of Python. Conda is available on OLCF systems, and loading the default Python module loads an Anaconda Python distribution. Loading this distribution automatically puts you in a “base” conda environment, which already includes packages that one can use for simulation, analysis, and machine learning.' Further down, it mentions: 'For users interested in using Python with Jupyter, see [Jupyter at OLCF](#) instead.' The next section is titled 'OLCF Python Guides' and contains a list of guides: 'Conda Basics Guide: Goes over the basic workflow and commands of Conda', 'Installing Parallel h5py Guide: Teaches you how to install parallel-enabled h5py and mpi4py', and 'Installing CuPy Guide: Teaches you how to install CuPy'.

← → ↻ 🔒 <https://docs.olcf.ornl.gov/software/python/index.html> 📄 ☆ 🔒 🔔 ☰

OAK RIDGE | LEADERSHIP
National Laboratory | COMPUTING
FACILITY

🏠 OLCF User Documentation

Search docs

New User Quick Start

- Accounts and Projects
- Connecting
- Systems
- Services and Applications
- Data Storage and Transfers

📁 **Software**

- Software News
- ML/DL & Data Analytics
- 📁 **Python on OLCF Systems**
 - Overview
 - OLCF Python Guides
 - Module Usage
 - How to Run
 - Best Practices
- Profiling Tools
- User-Managed Software
- Workflows
- E4S Software Stack
- Spack Environments
- Visualization Tools

» Software » Python on OLCF Systems

[Edit on GitHub](#) | [OLCF Home Page](#)

Python on OLCF Systems

Overview

In high-performance computing, [Python](#) is heavily used to analyze scientific data on the system. Some users require specific versions of Python or niche scientific packages to analyze their data, which may further depend on numerous other Python packages. Because of all the dependencies that some Python packages require, and all the types of data that exist, it can be quite troublesome to get different Python installations to “play nicely” with each other, especially on an HPC system where the system environment is complicated. [Conda](#), a package and virtual environment manager from the [Anaconda](#) distribution, helps alleviate these issues.

Conda allows users to easily install different versions of binary software packages and any required libraries appropriate for their computing platform. The versatility of conda allows a user to essentially build their own isolated Python environment, without having to worry about clashing dependencies and other system installations of Python. Conda is available on OLCF systems, and loading the default Python module loads an Anaconda Python distribution. Loading this distribution automatically puts you in a “base” conda environment, which already includes packages that one can use for simulation, analysis, and machine learning.

For users interested in using Python with Jupyter, see [Jupyter at OLCF](#) instead.

OLCF Python Guides

Below is a list of guides created for using Python on OLCF systems.

- [Conda Basics Guide](#): Goes over the basic workflow and commands of Conda
- [Installing Parallel h5py Guide](#): Teaches you how to install parallel-enabled h5py and mpi4py
- [Installing CuPy Guide](#): Teaches you how to install CuPy

Tools Overview. ML/DL & Data Analytics

The screenshot shows a web browser displaying the OLCF User Documentation website. The URL in the address bar is <https://docs.olcf.ornl.gov/software/analytics/index.html>. The page features a sidebar on the left with the Oak Ridge National Laboratory logo and a navigation menu. The main content area on the right is titled "ML/DL & Data Analytics" and contains a list of links to various machine learning and data analytics resources.

OLCF User Documentation

Search docs

- New User Quick Start
- Accounts and Projects
- Connecting
- Systems
- Services and Applications
- Data Storage and Transfers
- Software**
 - Software News
 - ML/DL & Data Analytics**
 - IBM Watson Machine Learning CE -> Open CE
 - R and pbdR on Summit
 - NVIDIA RAPIDS
 - Python on OLCF Systems
 - Profiling Tools
 - User-Managed Software
 - Workflows
 - E4S Software Stack
 - Spack Environments
 - Visualization Tools
- Training
- Contributing to these docs

ML/DL & Data Analytics

There are several options for various kinds of machine learning, deep learning, and data analytics tasks on OLCF systems.

- IBM Watson Machine Learning CE -> Open CE**
 - Getting Started
 - Running Distributed Deep Learning Jobs
 - Setting up Custom Environments
 - Best Distributed Deep Learning Performance
 - Example
- R and pbdR on Summit**
 - Loading R
 - How to Run an R Script
 - R Hello World Example
 - pbdR Hello World Example
 - Common R Packages for Parallelism
 - GPU Computing with R
 - More Information
- NVIDIA RAPIDS**
 - Overview
 - Getting Started
 - RAPIDS on Jupyter
 - RAPIDS on Summit
 - Setting up Custom Environments
 - BlazingSQL Distributed Execution

Jupyter at OLCF

← → ↻ https://docs.olcf.ornl.gov/services_and_applications/jupyter/index.html ☆ 🔒 📧 ☰

OAK RIDGE | LEADERSHIP
National Laboratory | COMPUTING
FACILITY

🏠 OLCF User Documentation

Search docs

- New User Quick Start
- ⊕ Accounts and Projects
- ⊕ Connecting
- ⊕ Systems
- ➡ **Services and Applications**
 - ⊕ Slate
 - ⊕ myOLCF
 - ⊖ **Jupyter**
 - ⊕ Overview
 - ⊕ Data Storage and Transfers
 - ⊕ Software
 - ⊕ Training
 - ⊕ Contributing to these docs

» Services and Applications » Jupyter

[Edit on GitHub](#) | [OLCF Home Page](#)

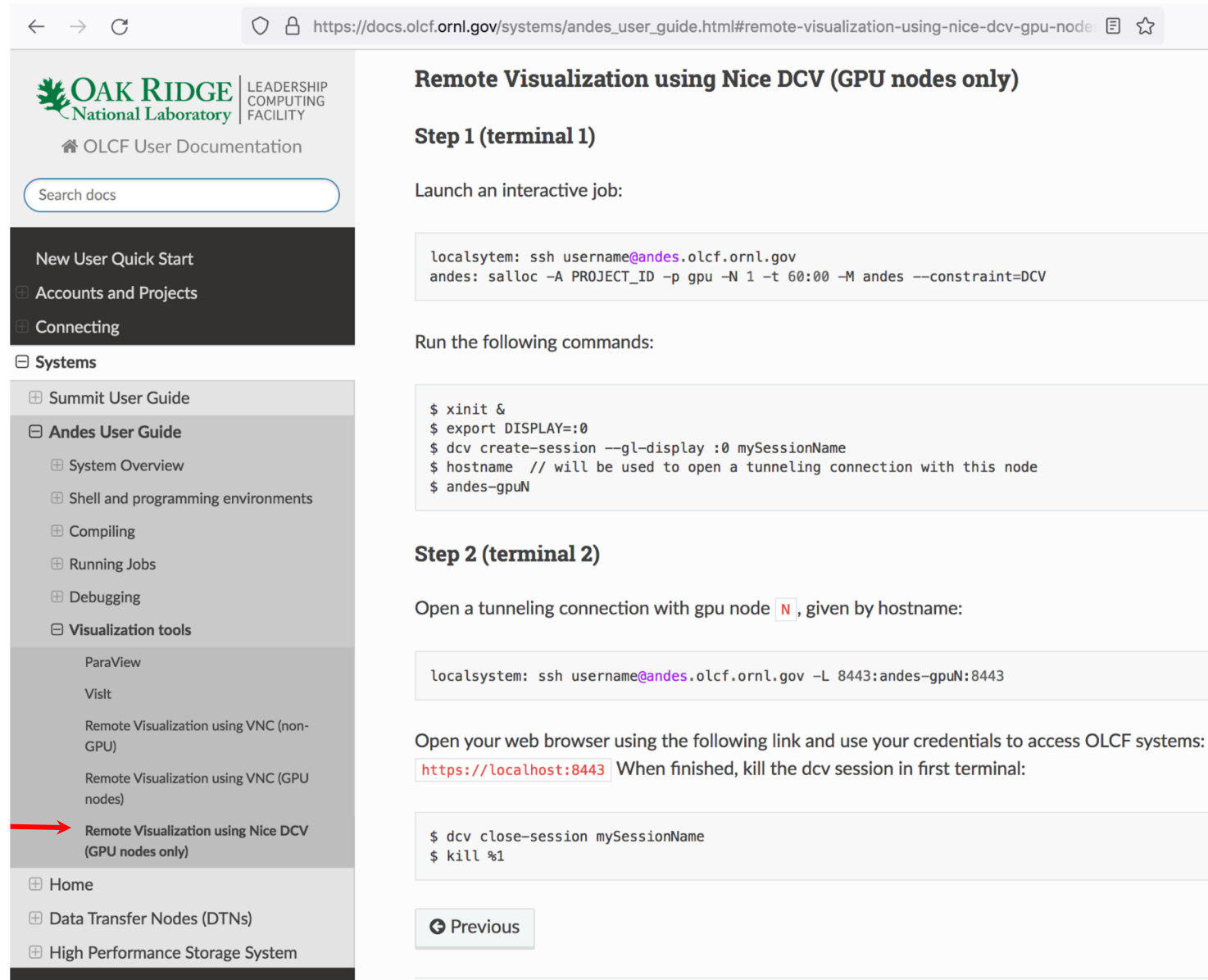
Jupyter

- Overview
 - [Jupyter at OLCF](#)
 - [Access](#)
 - [CPU vs. GPU JupyterLab \(Available Resources\)](#)
 - [Working within GPFS and NFS \(Launching a notebook\)](#)
 - [Conda environments and custom notebooks](#)
 - [Manually stopping your JupyterLab session](#)
 - [Things to be aware of](#)
 - [Example Jupyter Notebooks](#)

[← Previous](#) [Next →](#)

© Copyright 2021, OLCF.
Built with [Sphinx](#) using a [theme](#) provided by [Read the Docs](#).

Visualization tools



The screenshot shows a web browser displaying the OLCF User Documentation website. The left sidebar contains a navigation menu with the following items: "New User Quick Start", "Accounts and Projects", "Connecting", "Systems", "Summit User Guide", "Andes User Guide" (expanded), "System Overview", "Shell and programming environments", "Compiling", "Running Jobs", "Debugging", "Visualization tools" (expanded), "ParaView", "Visit", "Remote Visualization using VNC (non-GPU)", "Remote Visualization using VNC (GPU nodes)", "Remote Visualization using Nice DCV (GPU nodes only)" (highlighted with a red arrow), "Home", "Data Transfer Nodes (DTNs)", and "High Performance Storage System". The main content area is titled "Remote Visualization using Nice DCV (GPU nodes only)" and contains the following sections:

Remote Visualization using Nice DCV (GPU nodes only)

Step 1 (terminal 1)

Launch an interactive job:

```
localsystem: ssh username@andes.olcf.ornl.gov
andes: salloc -A PROJECT_ID -p gpu -N 1 -t 60:00 -M andes --constraint=DCV
```

Run the following commands:

```
$ xinit &
$ export DISPLAY=:0
$ dcv create-session --gl-display :0 mySessionName
$ hostname // will be used to open a tunneling connection with this node
$ andes-gpuN
```

Step 2 (terminal 2)

Open a tunneling connection with gpu node **N**, given by hostname:

```
localsystem: ssh username@andes.olcf.ornl.gov -L 8443:andes-gpuN:8443
```

Open your web browser using the following link and use your credentials to access OLCF systems:
<https://localhost:8443> When finished, kill the dcv session in first terminal:

```
$ dcv close-session mySessionName
$ kill %1
```

Previous

Remote Desktops. TurboVNC and NiceDCV

- Users can access and control a remote desktop (xfce) running on Andes.
 - Useful when a viz. tool does not have a client/server architecture as in Paraview or Visit
- TurboVNC and NiceDCV as a low-latency alternative to X-forwarding
 - RFB (remote frame buffer protocol) for optimal keyboard and mouse event and frame buffer delivery.
 - Low latency encoding
 - TurboVNC - JPEG encoding on CPU's SIMD instruction set
 - NiceDCV - H264 encoding using NVENC on GPU

Remote Desktops.

	TurboVNC	TurboVNC / VirtualGL	NiceDCV
Open source	Yes	Yes	No
Availability	Regular nodes	GPU nodes	GPU Nodes (5 seats)
Use case	Non graphics intensive apps, e.g. 2D graphics, any user interface (matlab, performance tools, editors, etc.)	GPU accelerated 3D graphics, apps with no client/ server architecture e.g. VMD, yt's 3D visualization, USC Chimera, custom viz. tools CUDA+OpenGL	GPU accelerated 3D graphics, apps with no client/ server architecture e.g. VMD, yt's 3D visualization, USC Chimera, custom viz. tools CUDA+OpenGL If TurboVNC/VirtualGL provide laggy response
Compression	TurboJPEG (SIMD) • Configurable Quality / Compression settings	TurboJPEG (SIMD) • Configurable Quality / Compression settings	H264 (NVENC) • Configurable Quality / Compression settings
Client	vncviewer	vncviewer	Web browser

Visualization Tools

The screenshot shows a web browser at the URL https://docs.olcf.ornl.gov/software/viz_tools/index.html. The page header includes the OAK RIDGE National Laboratory logo and the text 'LEADERSHIP COMPUTING FACILITY'. Below the header is a search bar labeled 'Search docs'. The left sidebar contains a navigation menu with the following items: 'New User Quick Start', 'Accounts and Projects', 'Connecting', 'Systems', 'Services and Applications', 'Data Storage and Transfers', 'Software' (expanded), 'Software News', 'ML/DL & Data Analytics', 'Python on OLCF Systems', 'Profiling Tools', 'User-Managed Software', 'Workflows', 'E4S Software Stack', 'Spack Environments', 'Visualization Tools' (highlighted with a red arrow), 'Visit', 'ParaView', 'Training', and 'Contributing to these docs'. The main content area is titled 'Visualization Tools' and contains two sections: 'Visit' and 'ParaView'. The 'Visit' section lists links: 'Overview', 'Installing and Setting Up Visit', 'Remote GUI Usage', 'Command Line Example', 'Troubleshooting', and 'Additional Resources'. The 'ParaView' section lists links: 'Overview', 'Installing and Setting Up ParaView', 'Remote GUI Usage', 'Command Line Example', 'Troubleshooting', and 'Additional Resources'. At the bottom of the main content area is a 'Previous' button. The footer contains the copyright notice '© Copyright 2021, OLCF.' and the text 'Built with Sphinx using a theme provided by Read the Docs.'

← → ↻ 🔒 https://docs.olcf.ornl.gov/software/viz_tools/index.html

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

OLCF User Documentation

Search docs

New User Quick Start

Accounts and Projects

Connecting

Systems

Services and Applications

Data Storage and Transfers

Software

Software News

ML/DL & Data Analytics

Python on OLCF Systems

Profiling Tools

User-Managed Software

Workflows

E4S Software Stack

Spack Environments

Visualization Tools

Visit

ParaView

Training

Contributing to these docs

» Software » Visualization Tools

Visualization Tools

- Visit
 - Overview
 - Installing and Setting Up Visit
 - Remote GUI Usage
 - Command Line Example
 - Troubleshooting
 - Additional Resources
- ParaView
 - Overview
 - Installing and Setting Up ParaView
 - Remote GUI Usage
 - Command Line Example
 - Troubleshooting
 - Additional Resources

Previous

© Copyright 2021, OLCF.

Built with Sphinx using a theme provided by [Read the Docs](#).

Thanks!

Issues and feedback

help@olcf.ornl.gov