
OSG Connect

Thursday 9:00 AM

Bala Desinghu (bm@uchicago.edu)

University of Chicago, OSG User Support Team

Overview



- What is OSG Connect
- OSG Connect Services
 - Help Desk and User Support
 - Software Support
 - Data Management: Storage and Transfer

Open Science Grid (OSG)



A **framework** for large scale distributed resource sharing addressing the technology, policy, and social requirements of sharing computing resources.

Integrates computing and storage resources from over 120 sites in the U.S. These resources are owned by virtual organizations (VOs)



A Virtual Organization (VO) is a set of groups or individuals defined by some common cyber-infrastructure need. This can be a scientific experiment, a university campus or a distributed research effort.

Open Science Grid (OSG)



- OSG supports a default virtual organization (VO) called "OSG"
- If you are not already part of a VO, you can join OSG via [OSG Connect](#)
- OSG welcomes any researcher affiliated with an U.S. institution!



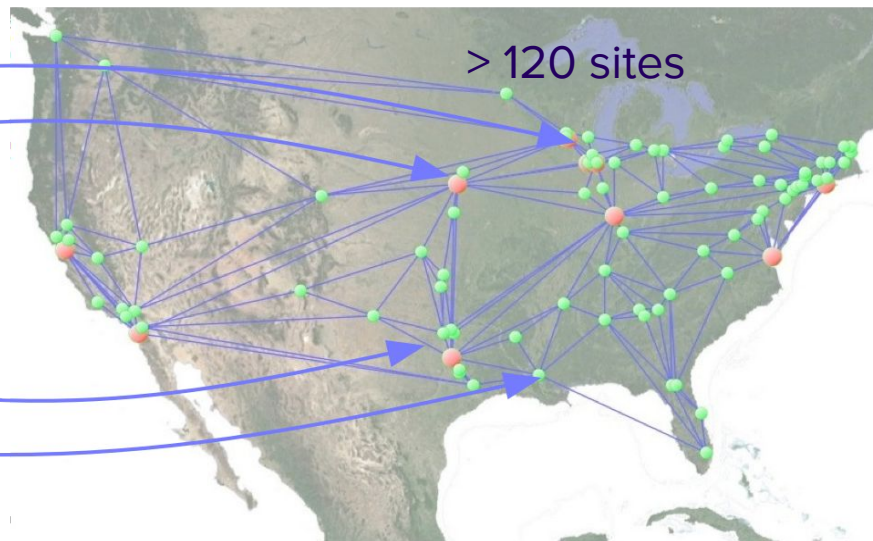
Tell colleagues who don't have a local VO!

Open Science Grid (OSG)



Local VO
(Campus or group specific)

OSG VO
(OSG Connect, XD Connect)



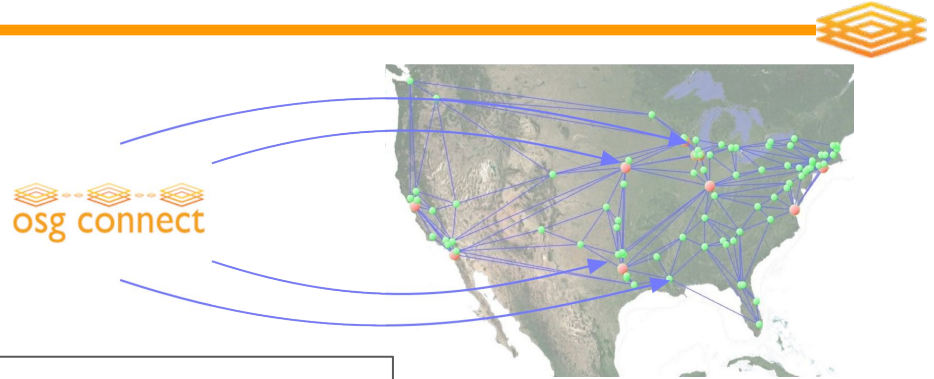
- Local VO: Users with Campus/Org
- OSG Connect: Affiliated with U.S. research university/institute
- XD Connect: Users with XSEDE allocations

Review: OSG Submit Locations



	Local	OSG Connect	XD Connect
Available to:	Researchers affiliated with institution	Affiliates of U.S. research orgs	Users with XSEDE allocation
Compute resource	Opportunistic + Allocations	Opportunistic	Allocations
Limit on CPU hrs	Unlikely	No	Yes (per allocation)
User support	Local staff	Help Desk, Documentation, Support chat, email, and tickets	OSG Connect Support and, if available, campus champion
Submit location	Local submit server	login.osgconnect.net (user-training.osgconnect.net)	xd-login.opensciencegrid.org

OSG Connect



- OSG VO
- Jobs are submitted via HTCondor
- Provides online guides, remote human support, software, and data support
- Submit locally, run globally

Overview



- What is OSG Connect
- OSG Connect Services
 - Help Desk and User Support
 - Software Support
 - Data Management: Storage and Transfer

Help Desk and User Support



- Knowledge Base
 - User guides/tutorials
 - HTC Recipes
- Forums
- “How do I...?” articles
- Interactive online chat
- Workshops and outreach
- Help Desk: <https://support.opensciencegrid.org>
- Support email: user-support@opensciencegrid.org

Help Desk Tickets



 **opensciencegrid** help desk Welcome
Login Sign up

Home Solutions

How can we help you today?

Enter your search term here...

SEARCH

 [New support ticket](#)

 [Check ticket status](#)

Create
Ticket

Knowledge base

Overview

Getting Started (4)

 [A welcome from the User Support Team](#)

 [Communicate with us via Twitter](#)

OSG XSEDE Users (1)

 [OSG XSEDE User Guide](#)

[Leave us a message!](#)

Online Chat

Help Desk Articles - Basic Topics



OSG Connect User Guide

Getting Started with OSG Connect (5)

- Registration and Login for OSG Connect
- OSG Connect Quickstart
- Start a Project with OSG Connect
- The "tutorial" Command
- Generate SSH key pair and add the public key to your account

Choosing Resources for Jobs (5)

- Steer your jobs with HTCondor job requirements
- Large Memory Jobs
- GPU Jobs
- Multicore Jobs
- Singularity Containers

Running applications on OSG Connect (6)

- Accessing Software using Distributed Environment Modules
- Software modules catalog
- Troubleshooting Condor errors
- Requesting a software installation
- Software transfer via HTCondor or HTTP

>> See all 6 articles

Getting started info and how-to's for serial HTC jobs, multicore jobs, containers, and more

Data Management

Introduction to data management on OSG (1)

- Guidelines for data management in OSG - Storage and Transfer

Data storage (1)

- Storage Solutions on OSG: home, local, scratch, stash, and public

Leave us a message!



Help Desk Articles - HTC Recipes



High Throughput Computing Recipes

Molecular Dynamics (4)

- A Simple NAMD Application
- NAMD with input data from Stash using HTTP
- Pegasus - NAMD example
- GROMACS

Quantum Chemistry (1)

- Electronic Structure Calculations with CP2K

Workflow solutions (5)

- Makeflow - Quickstart
- Makeflow - Running GROMACS simulations in sequence of steps
- DAGMan - NAMD example
- Pegasus
- Swift

R (3)

- Calculating Pi using R
- Adding external packages to your R jobs
- Scaling up compute resources

Using FreeSurfer on OSG (9)

- What's New in Fsurf 2.0

Drug Discovery (2)

- AutoDock Vina
- Managing AutoDock Vina Tasks with Pegasus

Bioinformatics (3)

- Sequence Search with BLAST
- Blast Similarity Search with Pegasus
- StashCache-Blast

MATLAB Runtime (4)

- Basics of compiled MATLAB applications - He
- RandomMatrix and Wigner's semi-circle law
- Parameter sweep: Resonance in a driven-damped harmonic oscillator
- Optimization Tool box: Simulated Annealing

Python (1)

- Virtualenv in Python

Machine Learning (1)

- TensorFlow

The tutorial command quickly prepares files to launch an HTC recipe.

There are a number of recipes for common workloads available.

Leave us a message!

OSG Connect Submit Hosts



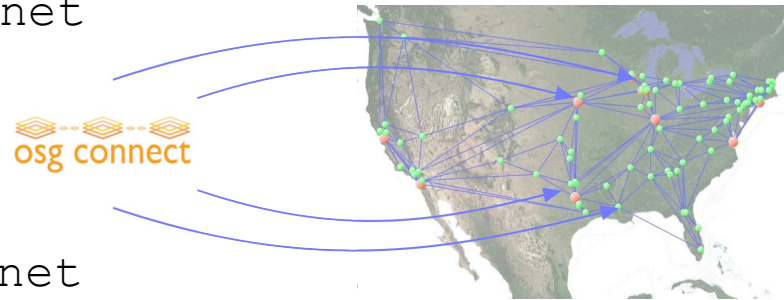
Submit Host for this workshop: `user-training.osgconnect.net`

For permanent accounts: `login.osgconnect.net`

- Today you need to be able to ssh to:

`username@user-training.osgconnect.net`

- Please let an instructor know if you are not able to log in.
- The workshop account is valid for two weeks. If you are interested in a long-term account, please signup after the workshop: <http://osgconnect.net/signup>



OSG Connect Submit Hosts



- Job Manager: HTCondor
- Workflow Managers: DAGMan, Pegasus, and, Makeflow
- Use `condor` commands and submit files as usual!
- One exception: `condor_status`

```
condor_status -pool osg-flock.grid.iu.edu
```

Tutorials on OSG Connect: `tutorial` Command



- Tutorials are maintained in [Github](#) and downloaded on demand
- Each tutorial's README is in the OSG Support site
 - <http://osg.link/connect/userguide>
 - <http://osg.link/connect/recipes>
- These are recommended for learning new techniques on OSG Connect

tutorial Command



```
sh$ tutorial
```

```
tutorial
```

```
usage: tutorial list           - show available tutorials
       tutorial info <tutorial-name> - show details of a tutorial
       tutorial <tutorial-name>     - set up a tutorial
```

Currently available tutorials:

```
AutoDockVina ..... Ligand-Receptor docking with AutoDock Vina
R ..... Estimate Pi using the R programming language
R-addlibSNA ..... Shows how to add R external libraries for the R
jobs
ScalingUp-Python ..... Python example to optimize a function on grid
points
```


tutorial Command



```
sh$ tutorial quickstart
```

```
Installing quickstart (master)...
```

```
Tutorial files installed in ./tutorial-quickstart.
```

```
Running setup in ./tutorial-quickstart...
```

```
sh$ cd tutorial-quickstart/
```

```
sh$ ls
```

```
Images      osg-template-job.submit  short.sh      tutorial02.submit
```

```
log  README.md      tutorial01.submit  tutorial03.submit
```

Overview



- What is OSG Connect
- OSG Connect Services
 - Help Desk and User Support
 - **Software Support**
 - Data Management: Storage and Transfer

Software Support



There are several ways to run software on OSG. (See Christina's talk yesterday). We will focus on two approaches.

- OASIS (**O**SG **A**pplication **S**oftware **I**nstallation **S**ervice)
- Singularity containers (Advanced material, time permitting)

What is OASIS?

- Repository for common user software
- Compiled and maintained by the user support team
- The repo contains about 180 software packages and libraries, including most commonly used open source science and engineering tools
- Available across ~90% of OSG sites
- Let us know if you need a package installed!



OASIS is easy to use



- Accessed with the `module` command
- Make sure the execution site has OASIS installed by including this in your HTCondor job description file:

```
requirements = (HAS_MODULES == true)
```

- In your execution script file:

```
module load package-name
```

Accessing OASIS on the Submit Host



```
sh$ module avail
----- /cvmfs/oasis.opensciencegrid.org/osg/modules/modulefiles/Core-----
ANTS/1.9.4          ectools          lapack/3.5.0          python/2.7          (D)
ANTS/2.1.0          (D)             eemt/0.1             lapack/3.6.1          (D)  python/3.4
MUMmer/3.23        elastix/2015     libXpm/3.5.10        python/3.5.2
OpenBUGS/3.2.3     entropy/2017.03.16  libgfortran/4.4.7    qhull/2012.1
R/3.1.1            (D)             espresso/5.1         libtiff/4.0.4        root/5.34-32-py34
R/3.2.0            (D)             espresso/5.2          (D)  llvm/3.6             root/5.34-32
R/3.2.1            (D)             ete2/2.3.8           llvm/3.7             root/6.06-02-py34 (D)
[...]
```

Basic OASIS Commands

- Load a software module:
`module load package-name`
- List loaded modules:
`module list`
- Unload a module (to prepare for another)
`module unload package-name`



Using OASIS on the Submit Host



```
sh$ module load R
```

```
sh$ which R
```

```
/cvmfs/oasis.opensciencegrid.org/osg/modules/R/3.1.1/bin/R
```

```
sh$ Rscript --version
```

```
R scripting front-end version 3.1.1 (2014-07-10)
```

```
sh$ module list
```

```
Currently Loaded Modules:
```

```
1) R/3.1.1
```


Overview



- What is OSG Connect
- OSG Connect Services
 - Help Desk and User Support
 - Software Support
 - **Data Management: Storage and Transfer**
(we cover some basics, more details in Derek's talk)

Data Storage on OSG Connect

System	Default Limit	Purpose	Network mounted	Backed Up
home	20 GB	Quick data access and not for submitting jobs	Yes	Yes
local-scratch	25 GB	Large temporary storage and I/O for your jobs. Files older than 30 days are automatically removed.	No	No
stash	200 GB	Large storage and accessible via Globus to/from your campus or laptop	Yes	No
public	10 GB	Sharing data and transfer input data via HTTP or stashcp	Yes	No

- Both **stash** and **public** are on the same filesystem.
- Public directory has an http interface and the files are world readable.
- Let us know if you need more!

Transferring Input Data for your Job

Method	Recommended File Size	Command	Purpose
HTCondor	< 100 MB	<code>transfer_input_files</code>	Input data from home, local-scratch, public or stash
HTTP and UNIX tools	< 1 GB	<code>wget</code> , <code>curl</code> , or <code>rsync</code>	Input data from <code>~/public</code> for HTTP tools (<code>wget</code> , <code>curl</code>), or home, local-scratch, public or stash (<code>rsync</code>)
StashCache	> 1 GB, < 50 GB	<code>stashcp</code>	Input data from <code>~/public</code>
GridFTP	> 1 GB < 50 GB	<code>gfal-copy</code>	Experts with large workflows. Contact us if you want to use it.

Submit Host

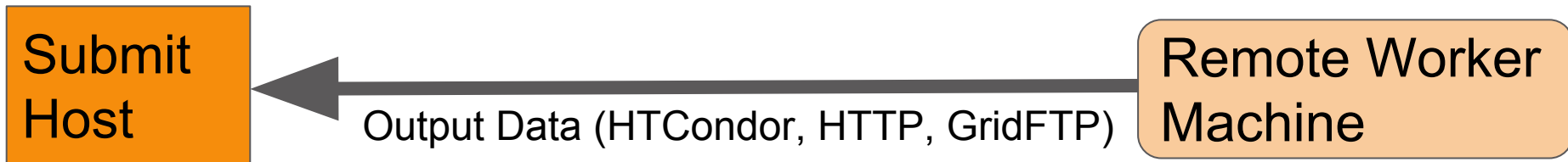
Input Data (HTCondor, HTTP, StashCache, etc.)

Remote Worker Machine

Transferring Output Data from your Job



Method	Recommended File Size	Command	Purpose
HTCondor	< 100 MB	<code>transfer_output_files</code>	Transfer data to submit directory
UNIX Tools	< 1 GB	<code>rsync</code> , <code>scp</code> , etc.	Transfer data to home, local-scratch, stash, etc.
GridFTP	> 1 GB, < 50 GB	<code>gfal_copy</code>	Experts with large workflows. Contact us if you want to use it.

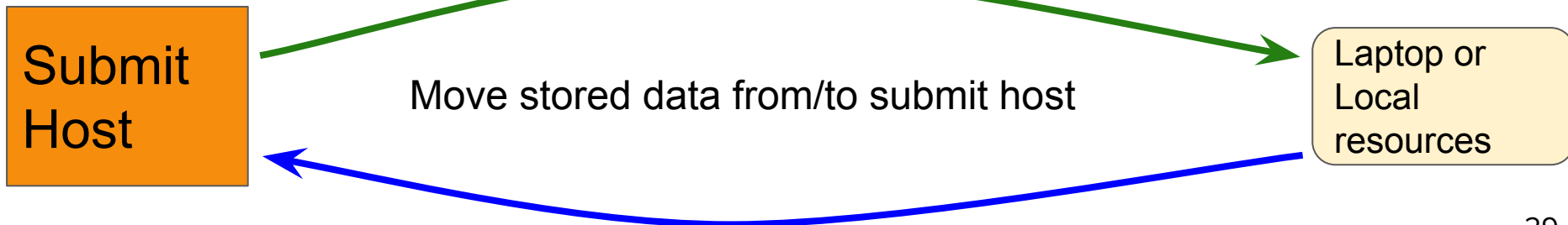


Data Transfer from OSG Connect



Method	Data Size	Tools
Secure Copy Protocol	< 1GB	scp, putty, WinSCP, gFTP, etc.
Globus	> 1GB	Globus web service or globus CLI

NOTE: Globus transfer is available through the OSG Connect Globus “endpoint”. You will need to a Globus personal endpoint to transfer to your laptop.



OSG Connect Exercises ([twiki](#))



```
ssh username@user-training.osgconnect.net
```

1.1 Get acquainted with OSG Connect

1.2 Do the “OSG Connect Quickstart”

1.3 Run ‘Gromacs’ via the OASIS module

1.4 Submit the tensorflow example “tf-matmul.py” on the OSG and see how it works.

BONUS: submit jobs from **osg-learn** using other software in OASIS (matlab, python, etc)

Advanced Material follows - Containers & TensorFlow

Singularity Containers



- Workload runs inside a container
- In fact, ~75% of OSG VO jobs run inside a container without the user even knowing about it
 - Provides a consistent environment and job separation
- User defined images are supported!
 - Complex software deployments
 - Non-Red-Hat-based Linux operating systems, e.g. Debian, Ubuntu, etc.



User created Singularity Containers



- You:
 - Define image in Docker
 - We have a set of images you can base your image on
 - Push image to Docker Hub
- Our system will:
 - Regularly check Docker Hub for updates on registered images
 - Automatically synchronize the latest version to CVMFS
 - Make the image available in exploded form under `/cvmfs/singularity.opensciencegrid.org/...`
 - This is a very efficient distribution mechanism for images

Singularity on OSG: Documentation



- Further details:
 - Help Desk article: <https://goo.gl/FmVkKN>
 - Derek's Blog: <https://goo.gl/LBtBbw>



A brief discussion about an example use case (TensorFlow)

Running TensorFlow Jobs using Singularity



TensorFlow

An open-source software library for Machine Intelligence

- TensorFlow Installation
 - TensorFlow is a very active project which requires up-to-date Python modules and system libraries - Makes it a difficult installation on long-term supported Red Hat Enterprise Linux distributions
- TensorFlow Singularity Solution - OSG provides vetted TensorFlow images
 - CPU version: directly imported from Docker image release by TensorFlow project
 - GPU version: based on NVIDIA's CUDA image, with TensorFlow added

Getting TensorFlow Tutorial



```
sh$ $ tutorial tf-matmul
```

```
Installing tf-matmul (master)...
```

```
Tutorial files installed in ./tutorial-tf-matmul.
```

```
Running setup in ./tutorial-tf-matmul...
```

```
sh$ cd ./tutorial-tf-matmul
```

```
sh$ ls
```

```
README.md  tf_matmul.py  tf_matmul.submit  tf_matmul_wrapper.sh
```

Using Singularity to run TensorFlow



Running the singularity container on the submit host

```
sh$ python tf_matmul.py
```

```
Traceback (most recent call last):
```

```
File "tf_matmul.py", line 3, in <module>
```

```
import tensorflow as tf
```

```
ImportError: No module named tensorflow
```

```
sh$ singularity shell /cvmfs/singularity.opensciencegrid.org/tensorflow/tensorflow:latest
```

```
sh$ python tf_matmul.py
```

```
result of matrix multiplication
```

```
=====
```

```
[[ 1.00000000e+00  0.00000000e+00]
```

```
 [ -4.76837158e-07  1.00000024e+00]]
```

```
=====
```

Throws error because tensorflow is only available in the container environment and not as a regular package.

Start a container and a shell inside the container

Inside the container, the job execution is successful.

Requesting TensorFlow Containers on OSG



Running on the remote worker machine

- Take a look at the job description file
- `Requirements = HAS_SINGULARITY == True` (Find a machine that has singularity installed)
- `+SingularityImage =`
`"/cvmfs/singularity.opensciencegrid.org/tensorflow/tensorflow:latest"`
(use the container image on cvmfs)

OSG Connect Exercises ([twiki](#))



```
ssh username@user-training.osgconnect.net
```

1.1 Get acquainted with OSG Connect

1.2 Do the “OSG Connect Quickstart”

1.3 Run ‘Gromacs’ via the OASIS module

1.4 Submit the tensorflow example “tf-matmul.py” on the OSG and see how it works.

BONUS: submit jobs from **osg-learn** using other software in OASIS (matlab, python, etc)