OSG Connect

Thursday 9:00 AM

Bala Desinghu (bala.desinghu@rutgers.edu)
Rutgers University, Office of Advanced Research Computing
University of Chicago, OSG User Support Team (former)
Overview

● What is OSG Connect

● OSG Connect Services
  ○ Help Desk and User Support
  ○ Software Support
  ○ Data Management: Storage and Transfer
The OSG facilitates access to distributed high throughput computing for research in the US.

Integrates computing and storage resources from about 100 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.
OSG Provides Opportunistic Resources

- OSG supports a default virtual organization (VO) called "OSG"
- **List of VO's**
- 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!_
OSG Connect: Entry point to access the OSG opportunistic resources

- Enables US researchers to access the OSG compute resources
- Jobs are submitted via HTCondor
- Provides online guides, remote human support, software, and data support
- Submit locally, run globally
Overview

● What is OSG Connect (Questions?)

● OSG Connect Services
  ○ Help Desk and User Support
  ○ Software Support
  ○ Data Management: Storage and Transfer
Help Desk: [https://support.opensciencegrid.org](https://support.opensciencegrid.org)

Create Ticket

Online Chat

Support email: user-support@opensciencegrid.org
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

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

Getting started info and how-to's for serial HTC jobs, multicore jobs, containers, and more
The tutorial command quickly prepares files to launch an HTC recipe.

There are a number of recipes for common workloads available.
Submit Host for this workshop: training.osgconnect.net

For permanent accounts: login.osgconnect.net (login01, login02, login03)

● Today you need to be able to ssh to: username@training.osgconnect.net
● Please let an instructor know if you are not able to log in.
● The workshop account is valid for a month. If you are interested in a long-term account, please signup: 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
flock.opensciencegrid.org
```
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/userguide)
  - [http://osg.link/connect/recipes](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 (Questions?)
  - **Software Support**
  - Data Management: Storage and Transfer
Software Support: OASIS

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

- OASIS (OSG Application Software Installation Service)
- Singularity containers (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!
## Accessing OASIS on the Submit Host

[training.osgconnect.net/login.osgconnect.net]

Use `module avail` to see the available packages in OASIS:

```bash
$ module avail
```

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Deps</th>
<th>(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANTS/1.9.4</td>
<td></td>
<td>ectools</td>
<td></td>
</tr>
<tr>
<td>ANTS/2.1.0</td>
<td>(D)</td>
<td>eemt/0.1</td>
<td></td>
</tr>
<tr>
<td>MUMmer/3.23</td>
<td></td>
<td>elastix/2015</td>
<td></td>
</tr>
<tr>
<td>OpenBUGS/3.2.3</td>
<td></td>
<td>entropy/2017.03.16</td>
<td></td>
</tr>
<tr>
<td>R/3.1.1</td>
<td>(D)</td>
<td>espresso/5.1</td>
<td></td>
</tr>
<tr>
<td>R/3.2.0</td>
<td></td>
<td>espresso/5.2</td>
<td>(D)</td>
</tr>
<tr>
<td>R/3.2.1</td>
<td></td>
<td>ete2/2.3.8</td>
<td></td>
</tr>
</tbody>
</table>

...
Accessing OASIS on the Submit Host
(training.osgconnect.net/login.osgconnect.net)

```
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
```
Accessing OASIS for your job

In your job description:

requirements = (HAS_MODULES =?= true)

In your execution script file:

module load package-name

Submit Host

Remote Worker Machine (OASIS available)

Remote Worker Machine (no OASIS)
Basic OASIS Commands

- Load a software module:
  \[\text{module load package-name}\]

- List loaded modules:
  \[\text{module list}\]

- Unload a module (to prepare for another):
  \[\text{module unload package-name}\]
Overview

● What is OSG Connect

● OSG Connect Services
  ○ Help Desk and User Support
  ○ Software Support (Questions?)
  ○ Data Management: Storage and Transfer
    (we cover some basics, more details in Derek’s talk)
# Data Storage on OSG Connect

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

- 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!
Data Transfer in OSG Connect

Submit Host

Remote Worker Machine

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

Job Output Data (HTCondor, S3, gridFTP, etc.)

Move stored data from/to submit host
(secured copy, HTTP, S3, globus)
## Transferring Input Data for your Job

<table>
<thead>
<tr>
<th>Method</th>
<th>Recommended File Size</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTCondor</td>
<td>&lt; 100 MB</td>
<td>transfer_input_files</td>
<td>Input data from home, local-scratch, public or stash</td>
</tr>
<tr>
<td>HTTP and UNIX tools</td>
<td>&lt; 1 GB</td>
<td>wget, curl, or rsync</td>
<td>Input data from ~/public for HTTP tools (wget, curl), or home, local-scratch, public or stash (rsync)</td>
</tr>
<tr>
<td>StashCache</td>
<td>&gt; 1 GB, &lt; 50 GB</td>
<td>stashcp</td>
<td>Input data from ~/public</td>
</tr>
</tbody>
</table>
| GridFTP              | > 1 GB  
< 50 GB          | gfal-copy           | Experts with large workflows. Contact us if you want to use it.        |
## Transferring Output Data from your Job

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

**Submit Host**

Remote Worker Machine

Output Data (HTCondor, HTTP, GridFTP)
Data Transfer from OSG Connect

<table>
<thead>
<tr>
<th>Method</th>
<th>Data Size</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Copy Protocol</td>
<td>&lt; 1GB</td>
<td><code>scp</code>, <code>putty</code>, <code>WinSCP</code>, <code>gFTP</code>, etc.</td>
</tr>
<tr>
<td>Globus</td>
<td>&gt; 1GB</td>
<td>Globus web service or <code>globus CLI</code></td>
</tr>
</tbody>
</table>

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

Submit Host

Move stored data from/to submit host

Laptop or Local resources
ssh username@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

BONUS: Try yesterday’s examples using OASIS (matlab, python, etc)

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

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

- OASIS (OSG Application Software Installation Service)
- Singularity containers
Software portability

- Build with compiler tools (make, cmake, etc.)
- OASIS - portable modules via CVMFS
- Containers (Dockers, Singularity, Rockers, etc.)
- Virtual Machines

How much to pack?

<table>
<thead>
<tr>
<th></th>
<th>Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Small (about 10 - 20 times) compared to VM</td>
</tr>
<tr>
<td>Speed</td>
<td>Starts in milliseconds</td>
</tr>
<tr>
<td>Overhead</td>
<td>negligible</td>
</tr>
</tbody>
</table>
Singularity:

- can run either a Docker or a Singularity image.
- does not need a daemon process to run an image
- can run workloads as MPI or OpenMP jobs

Singularity in OSG:

- available in 75% of OSG machines
- user defined images are supported
Software support on the OSG

<table>
<thead>
<tr>
<th>User</th>
<th>OSG support team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compilation</td>
<td>Builds the package on submit host or remote workers.</td>
</tr>
<tr>
<td></td>
<td>May help to resolve software dependency questions on local or remote workers.</td>
</tr>
<tr>
<td>OASIS</td>
<td>Sends a software installation request to OSG team.</td>
</tr>
<tr>
<td></td>
<td>Installs the requested package in OASIS which are available on the OSG (≈ 90% machines)</td>
</tr>
<tr>
<td>Container</td>
<td>Builds the required image on personal machine (laptop/desktop), publish on docker hub, and give the dockerID to the OSG team.</td>
</tr>
<tr>
<td></td>
<td>Distributes the images on the OSG (≈75% machines). Also, helps to build the image.</td>
</tr>
<tr>
<td>VM</td>
<td>(I have not seen any production runs)</td>
</tr>
<tr>
<td></td>
<td>May have some support for special cases on some sites</td>
</tr>
</tbody>
</table>
User created images

You

- Build image and send the docker id to OSG team

Laptop or Desktop

Build and publish

Docker Hub

- User can’t build on Submit Host.
- Build on a personal machine.
- Docker is recommended for building the image.

OSG Team

Synced

OSG Image Repo (similar to OASIS model)

Replicated

/cvmfs/singularity.opensciencegrid.org/…
Build image with docker: a simple example

What you need?

- Docker installed on your personal machine
- Create an account on docker hub (DOCKER_USER_ID)
- Docker file (FROM, RUN, etc.)
- Learn basic docker commands (build, push, pull, search, etc.)

```bash
sh$ cat Dockerfile
# Dockerfile - Simple example
FROM ubuntu:latest
MAINTAINER Bala "bala.desinghu@rutgers.edu"
RUN apt-get update
RUN apt-get install -y python python-pip wget
RUN pip install numpy
sh$ docker build -t my-python-image .
sh$ docker push $DOCKER_USER_ID/my-python-image
```
Singularity on OSG: Documentation

- Further details:
  - Help Desk articles:
    - 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 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
sh$ $ tutorial tensorflow-matmul

Installing tensorflow-matmul (master)...

Tutorial files installed in ./tutorial-tensorflow-matmul.

Running setup in ./tutorial-tensorflow-matmul...

sh$ cd ./tutorial-tensorflow-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/opensciencegrid/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/opensciencegrid/tensorflow:latest" (use the container image on cvmfs)
Overview

- What is OSG Connect

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

Questions?
OSG Connect Exercises (SchoolPage)

ssh username@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
BONUS: Try yesterday’s examples using OASIS (matlab, python, etc)
1.4 Submit the tensorflow example “tf-matmul.py” on the OSG and see how it works.
Thank You
## Review: OSG Submit Locations

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