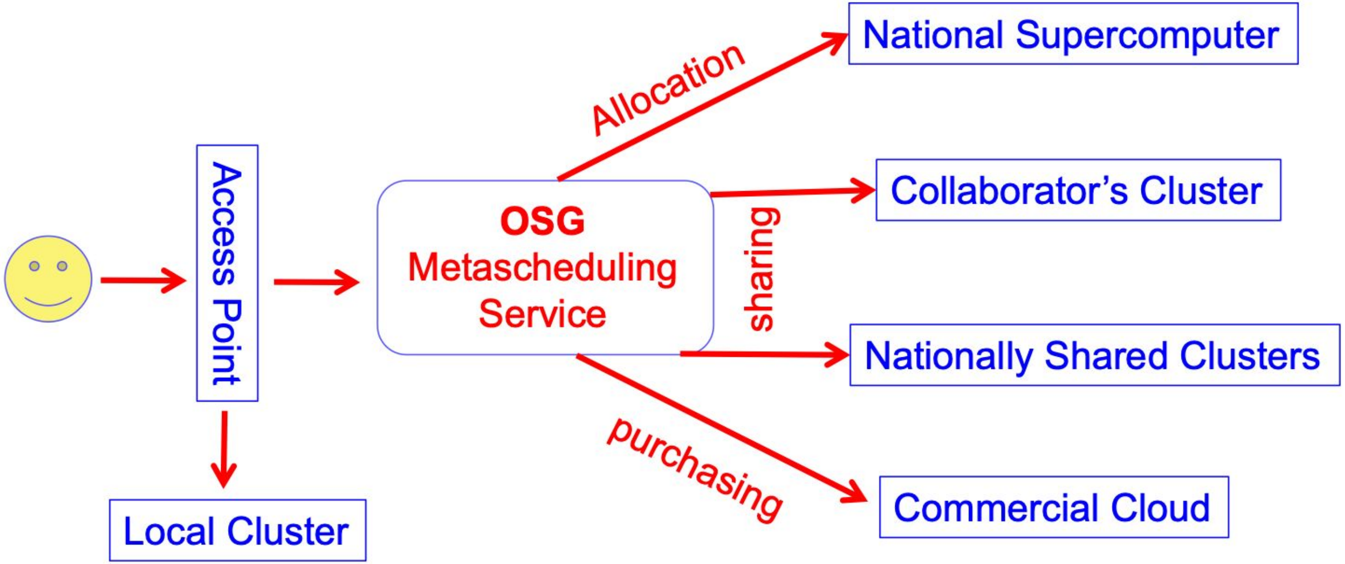
Bring Your Own Resources

Mats Rynge – OSG User Support
Eric Sedore – Syracuse University

Open Science Grid

OSG All Hands Meeting 2020


Access Point
OSG
Metascheduling
Service
Allocation
National Supercomputer
Collaborator's Cluster
sharing
Nationally Shared Clusters
purchasing
Commercial Cloud
Local Cluster

User Ownership

- Allocation
- Access
- Credit Card
- Policy

You scale it!

XSEDE Extreme Science and Engineering Discovery Environment

Campus Clusters

Jetstream

Chameleon

kubernetes

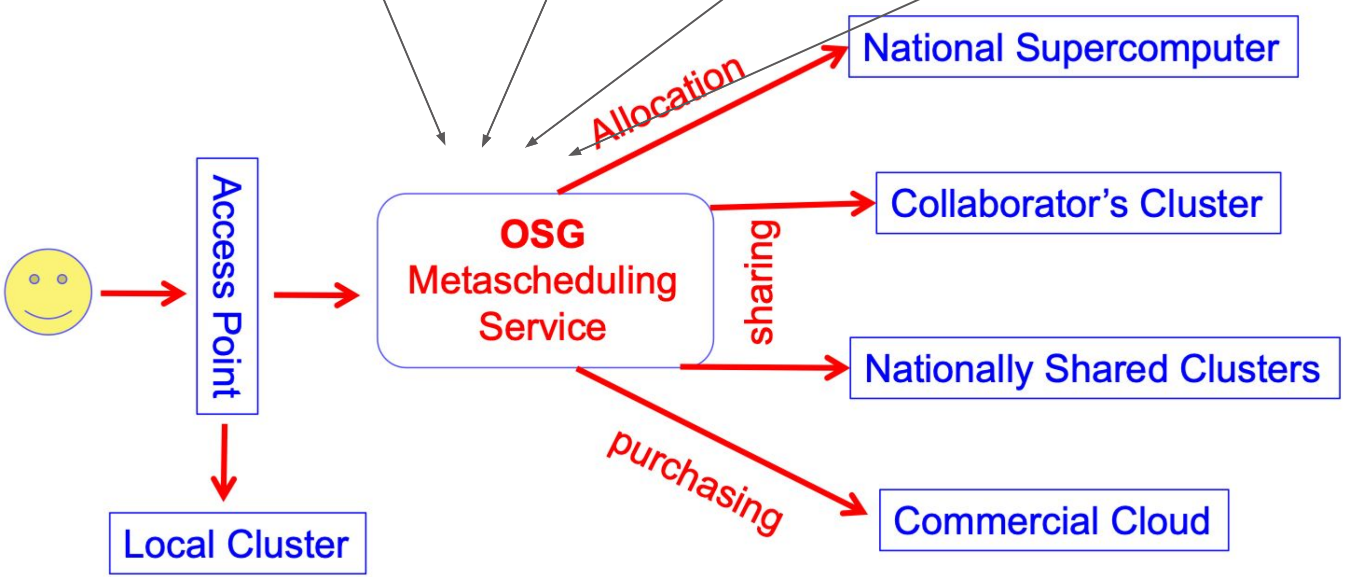
docker

aws

Azure

Google Cloud

OSG Infrastructure



Access Point

Local Cluster

OSG Metascheduling Service

Allocation

sharing

purchasing

National Supercomputer

Collaborator’s Cluster

Nationally Shared Clusters

Commercial Cloud

Motivations

1. Allocations on systems not “OSG-enabled”
    a. Policy prevents community accounts / individual PI accounts
    b. SSH 2 factor authentication

2. Scheduling on resources without traditional schedulers
    a. Clouds / k8s / desktops / …

3. Sometimes just a personal preference!

OSG infrastructure motivation: maintain one container which can be used on a wide variety of systems

Docker Container

CentOS 7 base

HTCondor 8.9.X (required for IDTOKENS)

Entrypoint to configure and start HTCondor

https://hub.docker.com/repository/docker/opensciencegrid/osgvo-docker-pilot

https://github.com/opensciencegrid/osgvo-docker-pilot

The enabler: IDTOKENS

New in HTCondor 8.9.x: Token-based authentication is a newer extension to PASSWORD authentication that allows the pool administrator to generate new, low-privilege tokens from a pool password. It also allows the administrator to install multiple passwords. As tokens are derived from a specific password, if an administrator removes the password from the directory specified in SEC_PASSWORD_DIRECTORY, then all derived tokens are immediately invalid.

Starting the container

1. Configure authentication. OSGVO administrators can provide the token, which you can then pass to the container via the TOKEN environment variable.

2. Set GLIDEIN_Site and GLIDEIN_ResourceName so that you get credit for the shared cycles.

3. Set the OSG_SQUID_LOCATION environment variable to the HTTP address to a valid Squid location.

4. Optional: Pick a directory where jobs can do I/O, and map it to /tmp inside with -v /somelocaldir:/tmp This is only required if you do not want the I/O inside the container instance.

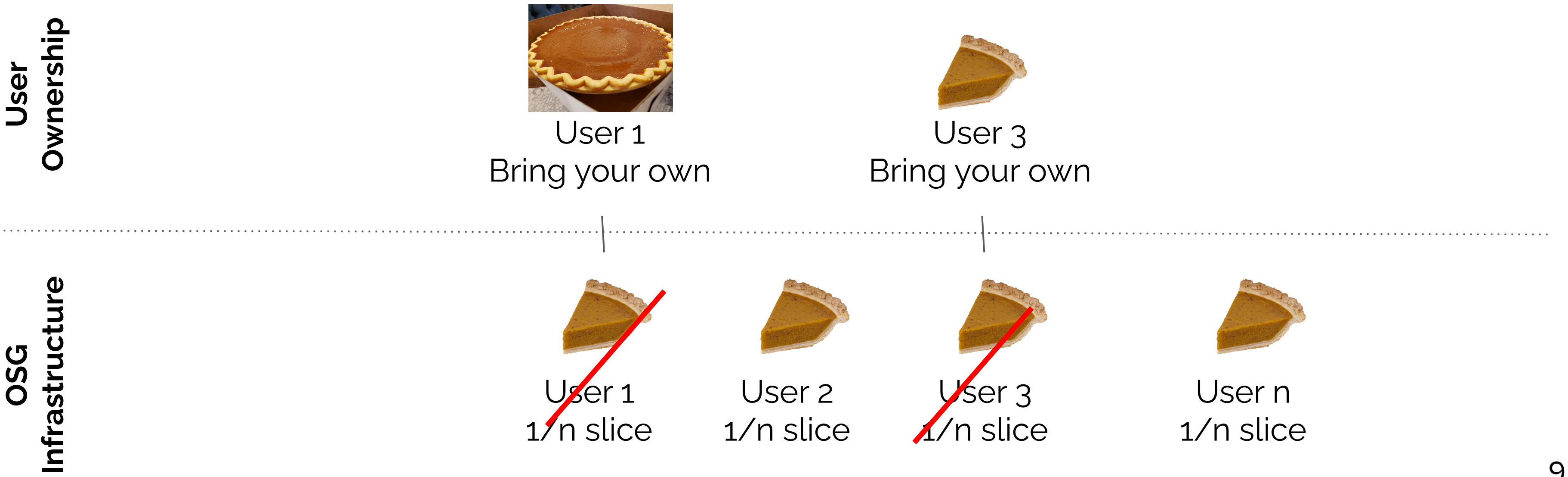
5. Optional: add to the START expression with GLIDEIN_Start_Extra. This is useful to limit the pilot to only run certain jobs.

Slurm example

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -t 48:00:00

export TOKEN="put_your_provided_token_here"

# Set this so that the OSG accounting knows where the jobs ran
export GLIDEIN_Site="SDSC"
export GLIDEIN_ResourceName="Comet"

# This is an important setting limiting what jobs your glideins will accept.
# At the minimum, the expression should limit the "Owner" of the jobs to
# whatever your username is on the OSG _submit_ side
export GLIDEIN_Start_Extra="Owner == \"rynge\""

module load singularity
singularity run --contain --bind /cvmfs docker://opensciencegrid/osgvo-docker-pilot
```

Fair share scheduling + your own

Bringing your own resources should not penalize your fair share part of the open pool



User Ownership

User 1
Bring your own

User 3
Bring your own

OSG Infrastructure

~~User 1~~
~~1/n slice~~

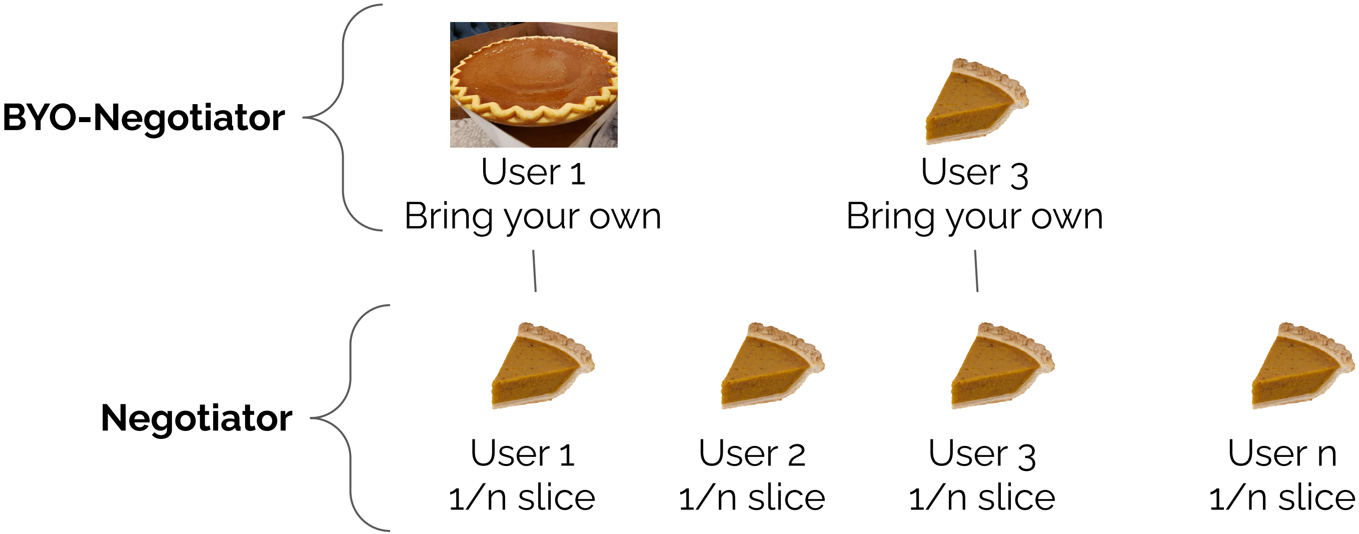
User 2
1/n slice

~~User 3~~
~~1/n slice~~

User n
1/n slice

Fair share scheduling + your own

One pool - two negotiators



BYO-Negotiator

User 1
Bring your own

User 3
Bring your own

Negotiator

User 1
1/n slice

User 2
1/n slice

User 3
1/n slice

User n
1/n slice

Syracuse University

Bring your own resources

(abstraction might help)

AHM OSG 2020

Information Technology Services

Abstraction for all workloads

Utilize the mobility and manageability of VMs /Containers to run OSG jobs on all of the University's research computing resources (servers, scavenged desktops, GPUs)

Simplify deployment of the OSG environment (versus trying to integrate with existing execute nodes)

OSG workloads fit in natural flow of workflow (VMs are deployed based on need)

Reduce security concerns - OSG VM does not run other workloads

Computational Resources @ Syracuse University

OrangeGrid - high throughput computing pool, scavenged desktop grid, 13,000 cores, 25TB of memory

Crush - compute focused cloud - 25,000 cores (50,000 slots with HT), 125TB of memory

SUrge - GPU focused compute cloud, 250 commodity NVidia GPUs

Why the drive to contribute?

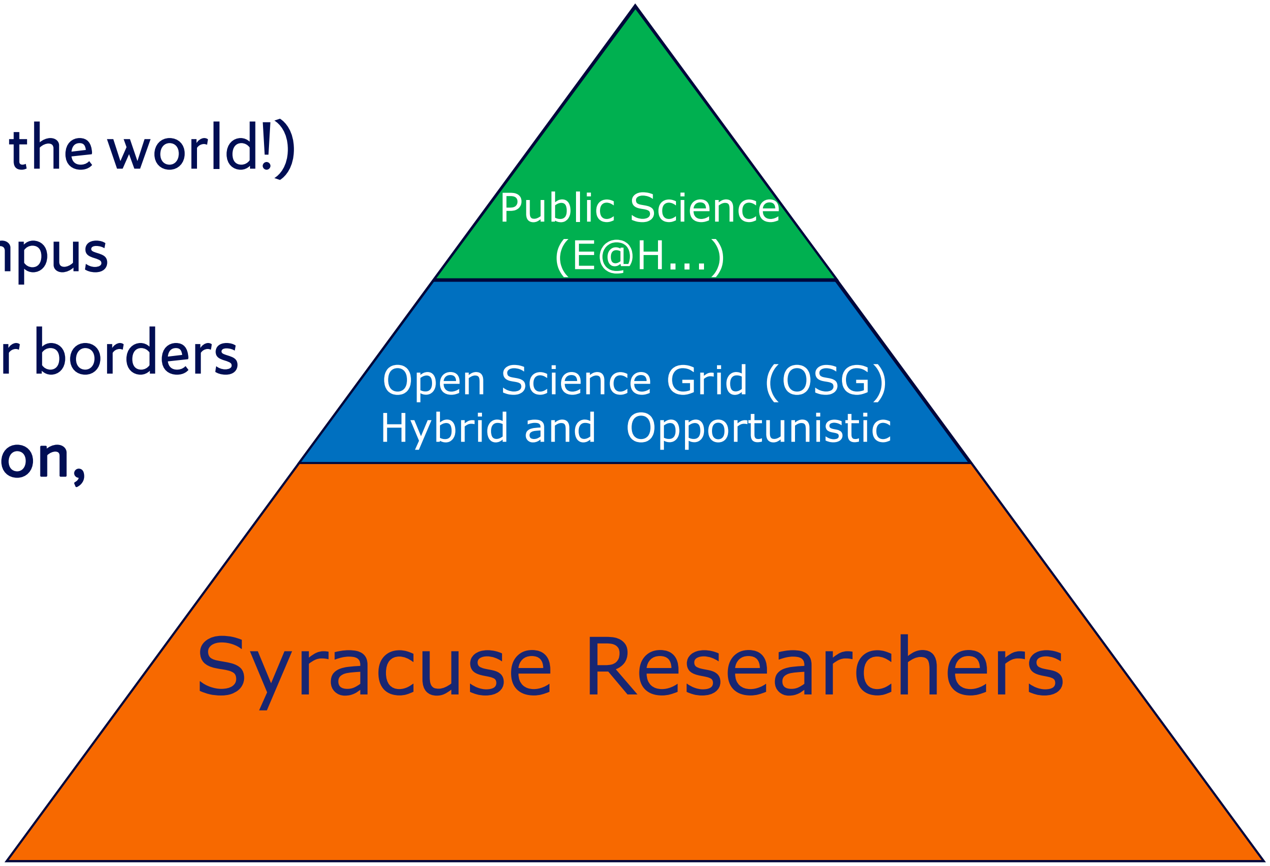
Advance Science (change the world!)

Serves researchers on campus

Allow for scale beyond our borders

Strive for 100% utilization,

100% of the time



Observations looking back

Investment in abstraction (a decision point years ago) allows simplified terraforming, scale, and workload management

Flexibility for research community has been an enormous advantage (yes, they can have it their way - well, within reason :-))

Assuage fears of intermixing internal and external workloads

Containers are a natural evolution for administrators and researchers (in fact we are running thousands of singularity instances in various pools today)