Building a shared and distributed computing facility for campus community - Rutgers initiatives and collaborations with the OSG

Balamurugan (Bala) Desinghu
Office of Advanced Research Computing
Rutgers University, NJ

Ongoing collaborations at Rutgers

OARC (Office of Advanced Research Computing) is young and strives to build cyber infrastructure, community, and collaborations

A few ongoing collaborations

- Eastern Regional Platform to enable multi-institute research projects
- Combined campus clusters to federate jobs and share data
- Integrating OSG services

The focus of this talk will be from a facilitator point of view on how to integrate OSG services with campus infrastructure so that the campus community can access vast computing resources.

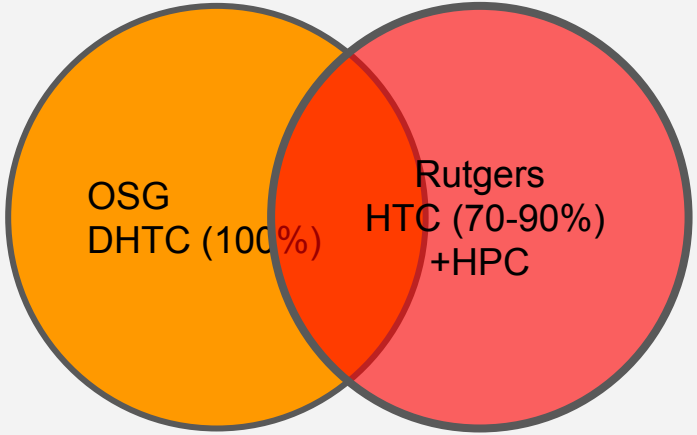
Resources on the OSG and Rutgers

If I want to help campus researchers utilizing OSG, I should learn the basics of the OSG and campus computing.

OSG	**Rutgers Campus**
National distributed high throughput computing (DHTC) for universities and institutions	Centralized cluster (s) for campus community
HTCondor	SLURM
Non-shared file system	Shared file system
Opportunistic cycles	Dedicated and Opportunistic cycles

This background knowledge is good, but usually not enough to support diverse users.

Computing styles on the OSG and Rutgers



OSG users do 100% of DHTC (distributed high throughput computing) while campus users do HTC (high throughput computing) and HPC (high performance computing).

The most interesting part is the overlap between DHTC and HTC. In principle, It is possible to gor from HPC to HTC to DHTC (HPC → HTC → DHTC). These transformations may be easy or hard. From campus support point of view, it is relatively easy to transform HTC → DHTC.

How to help the campus researchers to access OSG resources?

- First step is to find the users who have workloads suitable to run on the OSG
  - HTC → DHTC is easier compared to HPC → DHTC
  - Not HTC workloads working on a campus setting would be working on the OSG. For example, workloads requiring graphical interfaces won't work.

- OSG has several support channels, tools, solutions to assist DHTC workloads
  - Train the users and trainers (OSG UserSchool, OSG workshops like Quilt, Internet2, RMAC)
  - Send the users to Helpdesk
  - Build workflows using tools like Makeflow, Pegasus, etc., that can work on OSG and campus resources
  - Implement infrastructure solutions that can bridge OSG and Campus resources. Several possible solutions exist including CE, Hosted CE, BOSCO, PyGlideins, entry-points to OSG pool, etc…

Rutgers did some initial experiments with workflows, entry-points, storage, and PyGlideins.

Working with a traditional OSG user (3rd generation)

Prof. Joshua Plotkin
U Penn

1st gen

Prof. Oana Carja
Carnegie Mellon

Prof. Premal Shah
Rutgers

2nd gen

Tongji Xing (PhD student)
Rutgers
UserSchool18

3rd gen

next?

Strong supporters of the OSG

Workflow descriptions and requirements

Goal: Understand the role of biological noise in RNA transcription by evolutionary dynamics.

Workflow type: Scatter-gatherer type requiring data analysis at the end of several thousands of independent tasks.

Major resource requirements: For efficiency purpose, the tasks should be able to run on the OSG and campus clusters.

Based on user's research requirements, we decided to build workflows with Makeflow that can work on the OSG and all campus clusters.

Several successful workflows on the OSG and HPC centers.

Well known example: Pegasus workflow for LIGO Successfully executed on OSG,XSEDE, and other clusters.

Why users like meta schedulers? (Pegasus, Makeflow, Nextflow, Parsl, Swift, etc.)

- Scheduler agnostic
- Data dependencies are described in the workflow
- Can handle job dependencies
- Handle failures
- ...

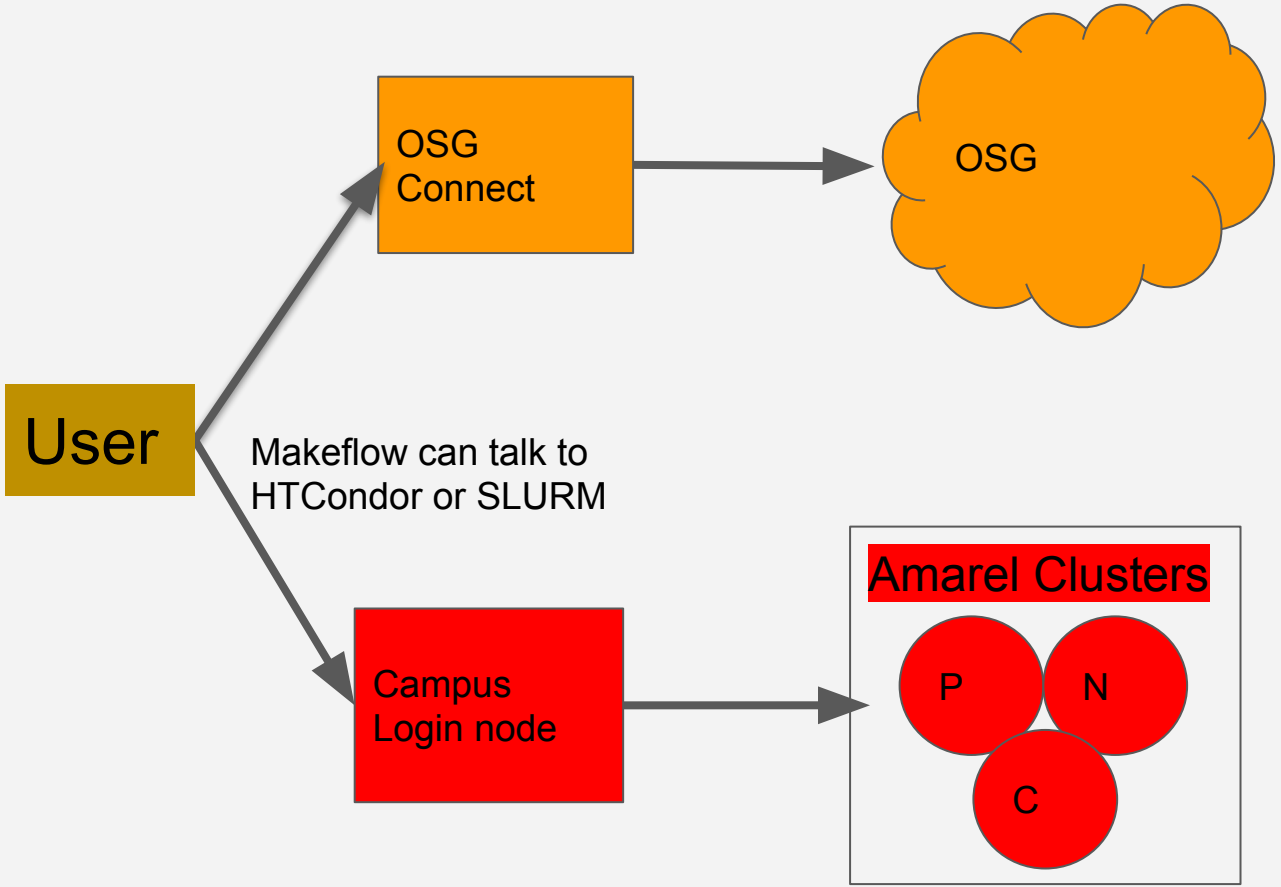
Works well for repeated workflows. Not good if the workload is a one time submission since some development time is involved.

Distributing workloads on the OSG and Campus Clusters

SSH into submit nodes and submit separate instances of makeflow.

The instances ran on OSG and Amarel Clusters.

Workflow remains the same. Only a small change in the argument specific to HTCondor and SLURM. nice!



Lessons learned from distributed workloads

Overall experience of the user was very positive. User liked a lot about distribution of workloads on multiple machines, automatic resubmission of failed jobs, not much to worry scheduler specifications, etc.

Some jobs ran for a long time or needed more memory that depend on on the input values. So, they repeatedly failed on the OSG.

- Increasing memory upon failure can be handed by pre-script (previously worked out for Rami's workflow with DAGMan)
- Long running jobs are still an issue. Currently, the user need to manually identify the failed jobs and execute them to campus cluster. An automated way will save a lot of time and work.

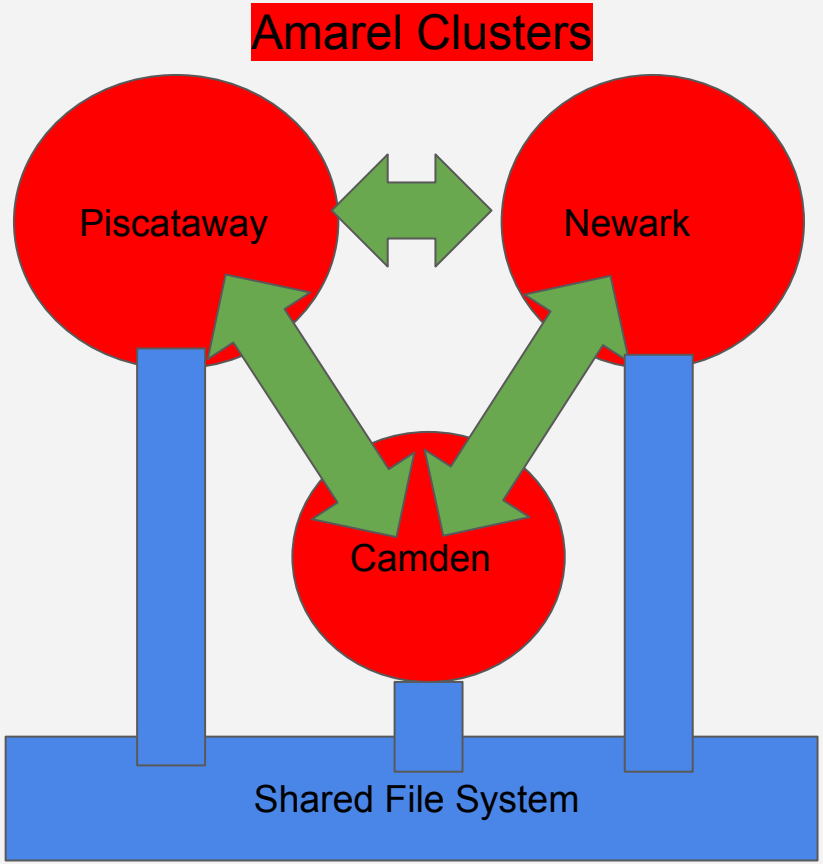
Instead of running several separate instances, it is good to have one instance of a running workflow that automatically splits the jobs across multiple clusters. Can we provide a common interface for the campus user to access all the available resources?

Shared and distributed computing facility at Rutgers

Job federation: Jobs can go anywhere (SLURM's cluster federation). Depending on the queue type, jobs are preemptable or non-premptable.

Data access: Home and Project filesets are accessible from all the clusters. Scratch is local to specific campus cluster.

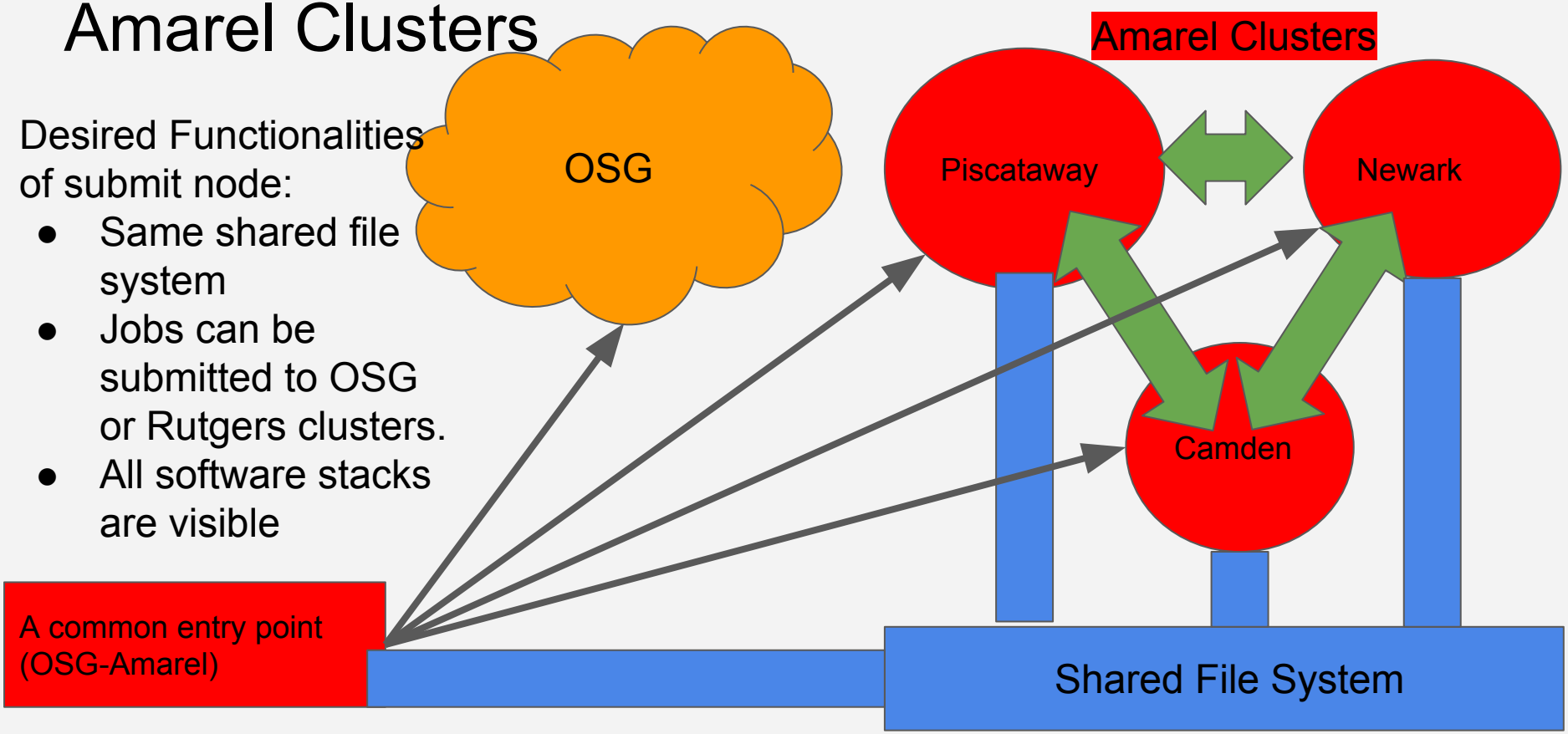
Campus users would be happy to work on an environment that unifies both OSG and campus resources.



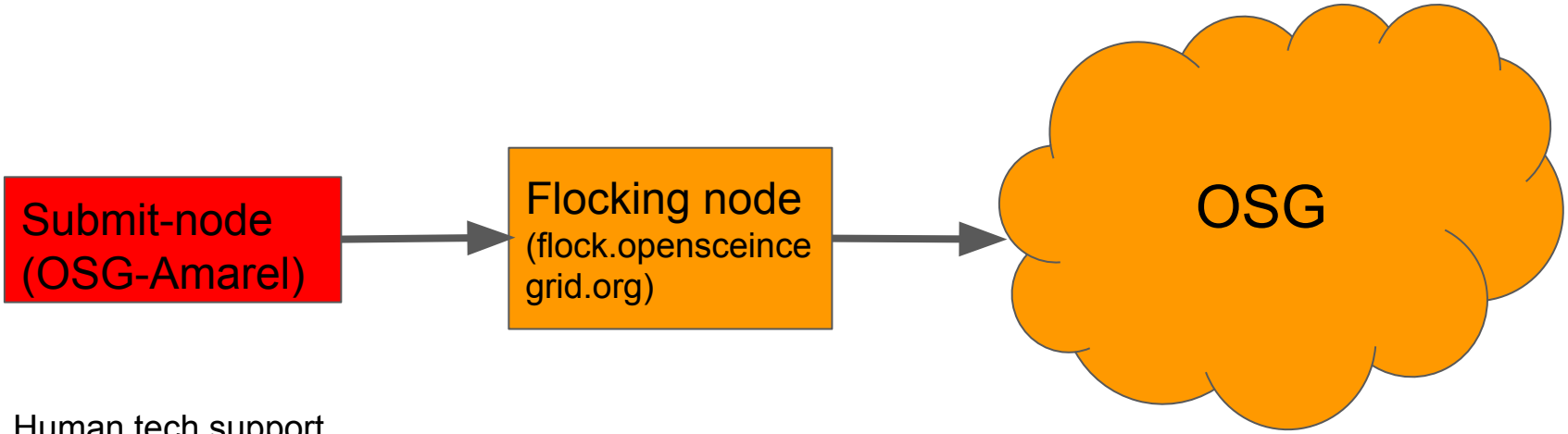
An common entry point to connect OSG and Amarel Clusters

Desired Functionalities of submit node:

- Same shared file system
- Jobs can be submitted to OSG or Rutgers clusters.
- All software stacks are visible



An entry point to OSG - OSG Amarel



Human tech support

Mats Rynge helped with the whole set up. Kevin helped on the Rutgers side. OSG team helped with initial questions and consultations.

Connecting to the outside world

Rutger clusters are behind firewall. Set up a login node that can open the ports to the outside world

Authentication: Pool password worked. Spend a few days in figuring out certificates.

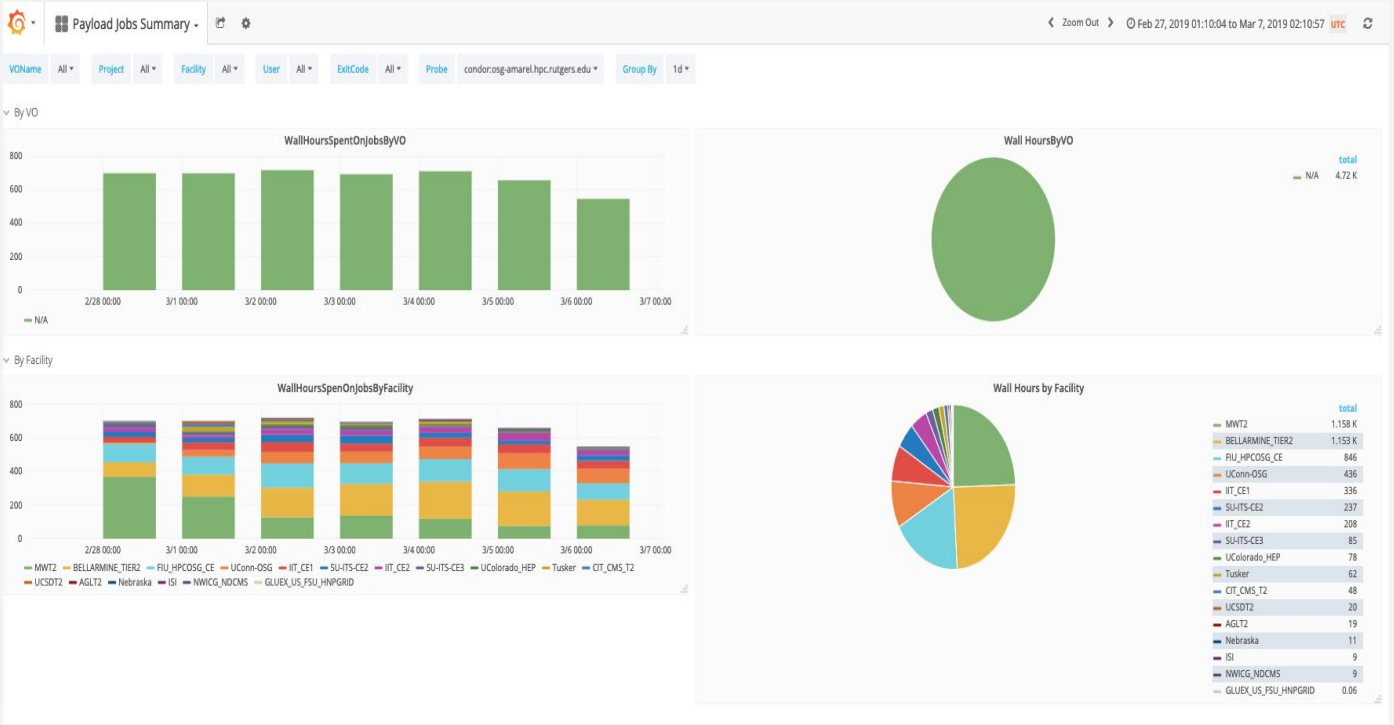
Software stacks

Installed CVMFS

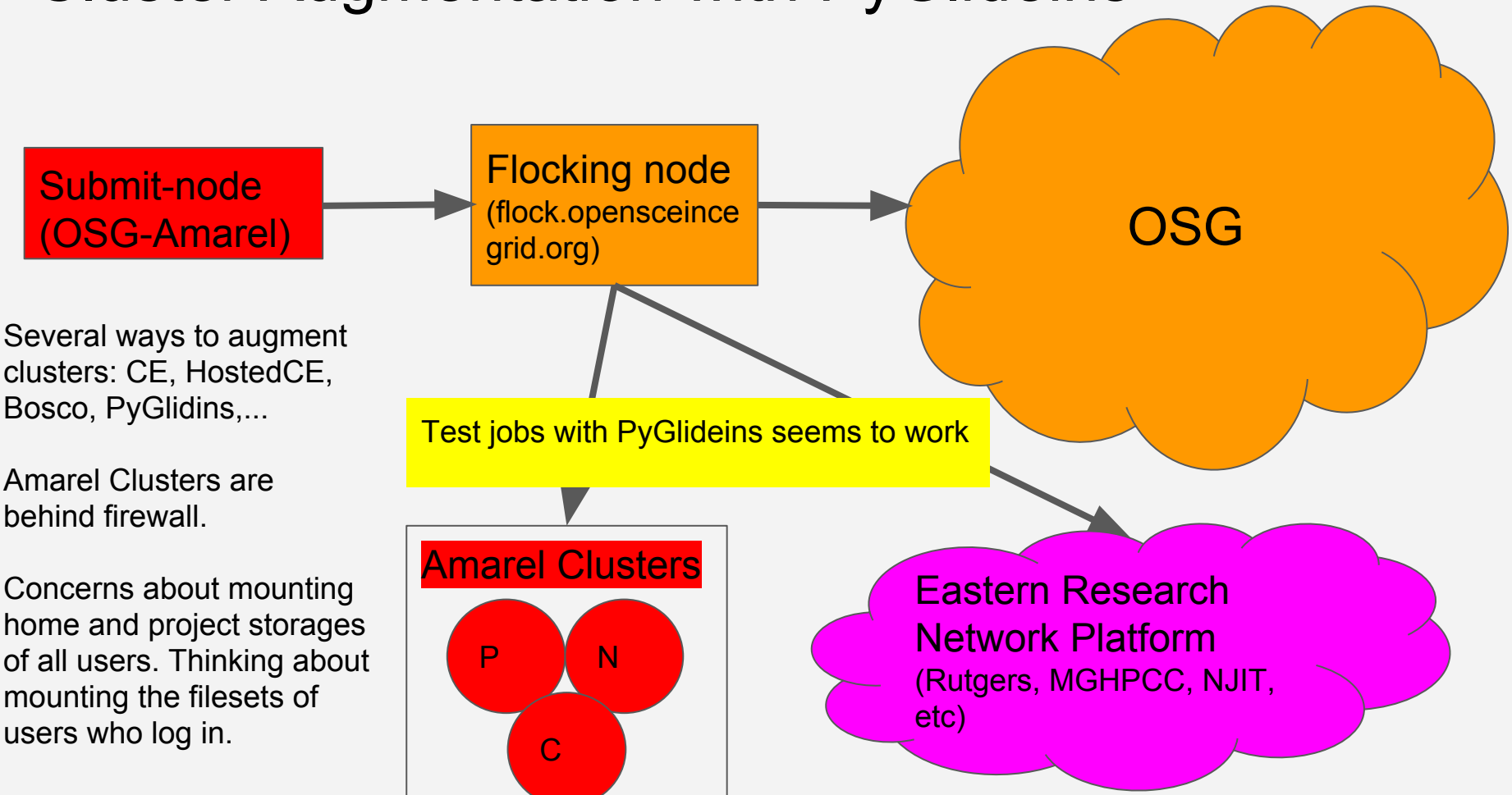
Gratia Accounting

Worked for some time and stopped working. Needs further fixing.

Jobs submitted on OSG-Amarel completed successfully on the OSG



Cluster Augmentation with PyGlideins

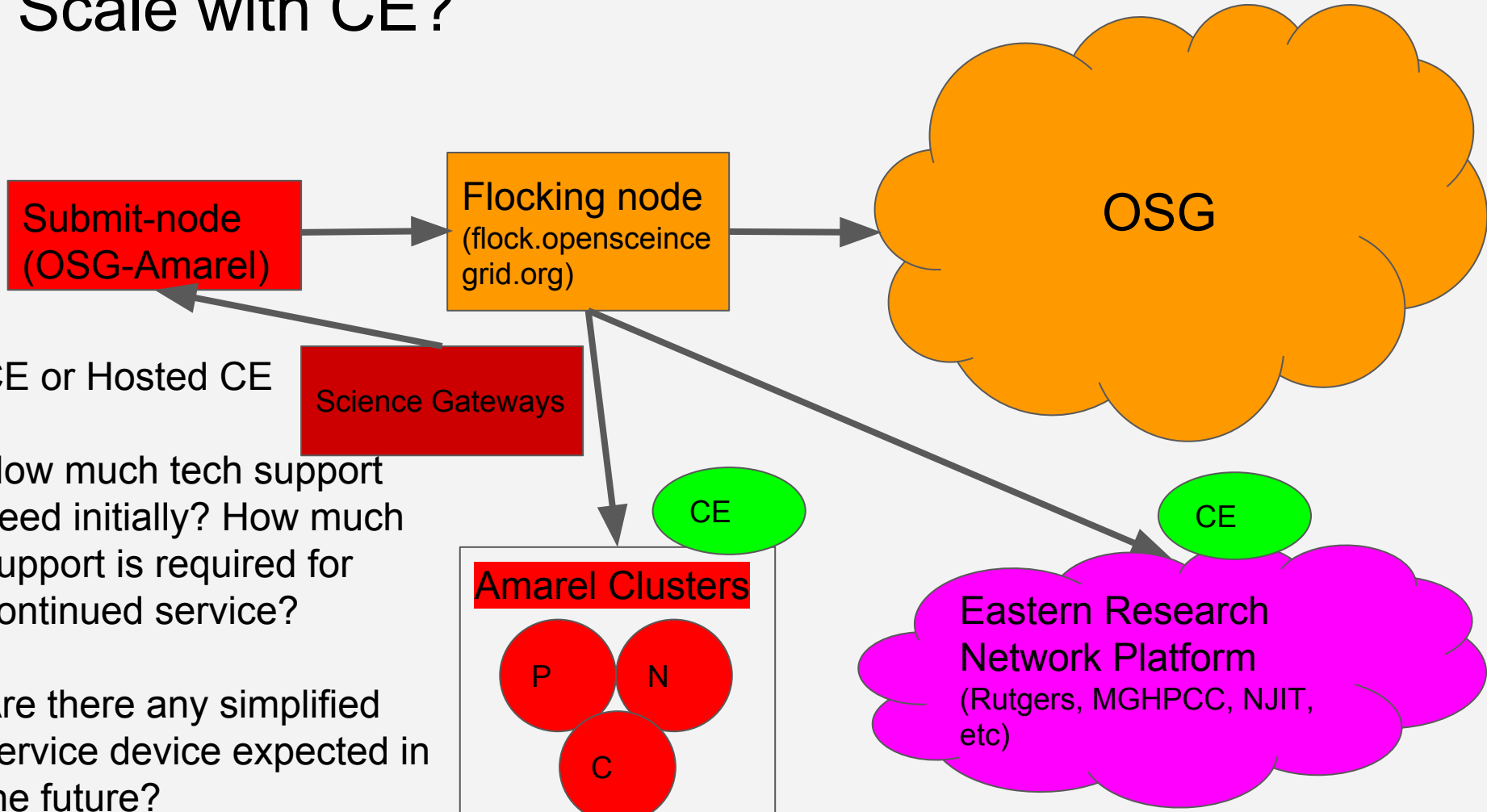


Several ways to augment clusters: CE, HostedCE, Bosco, PyGlidins,...

Amarel Clusters are behind firewall.

Concerns about mounting home and project storages of all users. Thinking about mounting the filesets of users who log in.

Scale with CE?



CE or Hosted CE

How much tech support need initially? How much support is required for continued service?

Are there any simplified service device expected in the future?

Summary

User landscape at Rutgers
- Diverse and distributed users
- Demand for HTC and HPC compute styles
- Strong need for data availability

Meta schedulers like Makeflow, Nextflow, Swift, Parsl, and Pegasus can help research groups to run their workloads on multiple computing resources. Auto-split the workloads with Pegasus or Makeflow+WorkQueue.

Cluster augmentation combined with extended storage support would help campus community.

Rutgers initiatives and collaborations with Eastern Research Platform and the OSG would help researchers across multiple institutions.

Acknowledgements

Dr. James Barr von Oehsen
Associate Vice President
Office of Advanced
Research Computing
Rutgers

Mats Rynge
OSG Team
University of Southern
California

Lincoln Bryant
UChicago Team

Dr. Galen Collier,
ACI-REF Team
Office of Advanced
Research Computing, ,
Rutgers

Kvein Abbey
System Administrato
Office of Advanced
Research Computing
Rutgers

Many thanks to many OSG team members for timely help on multiple occasions.

Thank You