OSG as an agile computing environment

Frank Würthwein
OSG Executive Director
UCSD/SDSC

March 3rd 2021
**Session dedicated to BYOR**

BYOR = Bring Your Own Resource

<table>
<thead>
<tr>
<th>Topic</th>
<th>Speaker</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSG as an agile computing environment</td>
<td>Frank Wuerthwein</td>
<td>11:00 - 11:25</td>
</tr>
<tr>
<td>CLAS12 simulations on the OSG</td>
<td>Dr Maurizio Ungaro</td>
<td>11:25 - 11:50</td>
</tr>
<tr>
<td>Accessing XSEDE allocations from the OSG access points</td>
<td>Mats Rynge</td>
<td>11:50 - 12:05</td>
</tr>
<tr>
<td>Searching for dark matter with XENON, OSG, and XD/XSEDE</td>
<td>Evan Shockley</td>
<td>12:05 - 12:30</td>
</tr>
</tbody>
</table>

We wanted to highlight some examples for BYOR, and encourage people to grow their resource pool via BYOR.
This talk is part aspirational and part report of what (sort of) exists as options to built an agile computing environment that includes OSG

Let’s start with an idea from my Monday talk ….
OSG Supports you to build your own dHTC environment

OSG supports a **modular software stack** and a “**Fabric of Services**” that allows organizations to create their own dHTC environment.
A Feature-Complete dHTC Environment

• **Compute Resource Pool**
  - Submission infrastructure that functions as compute “Access Point”
  - Workload management system
    - Complex workflows across heterogenous resources possible.
      - Easy to run workflows comprised of 100,000 jobs or more with complex dependencies between sets of jobs (full support of arbitrary DAGs).
  - Homogeneous runtime environment across heterogeneous resources

• **Data Resources**
  - Storage that functions as “Data Entry Point”
    - Origin to Caching infrastructure
    - Storage for data input to and output from workflows.
  - Transparent “Data Access” via Caching Infrastructure

How can we make this agile?
Agile means dynamic BYOR

• We want to support dynamic integration of resources that you bring to your dHTC environment that we/you have assembled for you with OSG services.
  – You buy resources in the commercial cloud that dynamically appear in your dHTC environment
  – You receive HPC allocations that you use from your dHTC environment
  – Your collaborator gives you resources that you use form your dHTC environment.

We certainly support BYOR (see following talks) … how dynamic can it be?

BYOR = Bring Your Own Resource
Agile growth of Compute Resources

• Adding your allocation to an existing Hosted-CE at an NSF HPC system

• Have your collaborator run our container/VM on their cloud infrastructure to connect to your pool.

• Use “glidein in a vacuum” from the login of your collaborators cluster

• Use HTCondor Annex to access AWS

• Use our VM on Google or Azure to connect to your pool.

• Use K8S integration to add Google TPUs to your pool.

It is possible to think outside the box to be agile!
Agility through K8S

• OSG operates K8S infrastructures at U.Chicago and UW-Madison for service deployments.
  – K8S native service deployments
  – OSG GitHub driven deployments
  – SLATE federation
  – K8S native federations (Admiralty & KubeFed)

• OSG integrates with K8S compute infrastructures at SDSC and U.Chicago
  – We have federated with K8S in the cloud to run applications on TPUs as a proof of principle.

• NSF offers an HPC system that supports K8S (Expanse)

Not clear all that’s possible makes sense to do. We are open to discuss unusual use cases.
Summary & Conclusion

• OSG is committed to support you to build your own dHTC environment.

• We encourage you to BYOR

• We are curious to engage with you on novel integrations of resources into agile dHTC environments.
  – Are there other dimensions of agile that we missed in this talk, and should be considering?

• Disclaimer: not everything that can be done makes sense to do.

  support@opensciencegrid.org
Acknowledgements

• This work was partially supported by the NSF grants OAC-2030508, OAC-1841530, OAC-1836650, and MPS-1148698