

OSG Perspective on Managing Software Stacks

Brian Bockelman

HSF Workshop, 22 January 2015

About OSG

- Open Science Grid. **Advance the science of distributed high-throughput computing (DHTC).**
- The OSG has been distributing a software stack for approximately a decade.
 - Right now, this is a set of RPMs in a yum repository for multiple platforms; these are integrated to provide a functional service(s).
 - In addition to the software stack, we offer a variety of training, security, operations, and consulting services.
- The OSG has a governing council which sets the strategic directions. These are executed by the OSG project, overseen by the OSG Executive Team.
 - The OSG project is jointly funded by DOE/NSF; implies we have a set of deliverables and a fixed effort profile.
 - The OSG Council provides a governing umbrella beyond the lifetime of any one project.
- The OSG Production Grid is a fabric of services for DHTC; includes many WLCG pledged sites, HEP sites, and non-HEP sites. Sharing (opportunistic use) is encouraged as a mechanism for providing value to all users of the production grid.
- I'd like to share our philosophy and some lessons learned. *Note* that this presentation focuses on the software stack and governance - not operations, security, user support, etc.

It's all about the Stakeholders

- The OSG is **stakeholder driven**. At the end of the day, they set priorities and requirements.
 - Note stakeholder driven may not necessarily imply stakeholders dictate all aspects of technology decisions.
 - This is an important fact when stakeholders have conflicting requirements - or when we disagree about technical approaches. Implementation is often a negotiation!
 - One way the OSG creates value is providing the software stack which allows the stakeholder to effectively run their computing resources.
 - This is critical for stakeholders to deliver physics.
- Without stakeholders - someone who depends critically on your existence - a project like OSG has a short lifespan!

What's left at the end?

- *How else* does OSG create value? I like to perform the mental exercise of “**what's left after the end of the funded project?**”
 - If we turned off all OSG-related services, what's left? How do we maximize this value?
- Here are my top answers:
 - Through our consulting, training, and evangelism, we have trained organizations on improving their use of DHTC. OSG's goal is advancing DHTC; we can make progress *even if our software isn't used*.
 - Provide contributions and knowledge to upstream software providers. We “win” whenever our patches or packaging is transitioned to upstream. We maximize this through constant collaboration with our upstream providers.
- By this metric, writing OSG-specific software provides little value!
 - Unless there's a very direct stakeholder need, **OSG does not write software**. We do submit patches and features to upstream as needed!
 - When we do write software, we tend toward having *thin layers* on existing software.

Managing the orphanage

- Another important concept is the “software orphanage”.
 - This is the set of software that has been abandoned by its developers but critical to our software stack.
 - For software in this state, we provide core maintenance and support. This effort is provided as a backstop to keep software in production.
 - Equally important, we help stakeholders reduce/eliminate usage of this software.
- Due to the required maintenance effort, each entry of the orphanage is a significant headwind for the OSG project.
 - For each active project, have a “living will” for migrating away from it.
 - Make sure there is effort and expertise available to execute the “living will”.
- Given enough time, all software will go into the orphanage: it is essential to have a mechanism to retire projects.

Keeping your nose clean

- Many others have commented on the importance of good software practices.
 - Automated internal unit & integration testing. We tend to only work on integration tests in OSG and rely on individual pieces to perform unit tests.
 - Keeping support for multiple platforms.
 - Keep track of the boring stuff - licenses, copyright, documentation, etc.
- Do all these! In addition, I suggest:
 - Collaborate with upstream software so they know what is important to our stakeholders and *what features they use*.
 - Make sure the software works with the base OS and the other software we ship.
 - The more upstream does this, the less work we have. The less work we have here, the more flexibility we have to apply our effort on other priorities.

Concluding thoughts

- Governance is as important.
 - The split between the OSG Council and OSG Project is important.
- Know what your organization is good at!
 - For OSG, this is DHTC, services, and a software stack. *Not* writing software.
- Know how you create value!
 - Advancing DHTC, stakeholder support, upstream contributions, enabling opportunistic use.
- Have a plan for keeping within the effort.
 - For us, it is managing the size of the “software orphanage”.
- Questions?