Bringing Your Own Capacity

Mátyás Selmeci and Todd L Miller

Contents

- Introduction and context
- Initial experiments in bringing capacity from HPC systems to the Open Science Pool
- HPC Annex and demonstration

Introduction

- “Bringing your own capacity” is a description of something we see happening in high-throughput computing.
- Researchers increasingly want to use compute capacity to which their HTCondor administrator does not have access.
- … from their usual access points, for their usual jobs.

Context: History

- First Phase:
  - condor_annex (self-service cloud bursting tool)
- Second Phase:
  - split-starter/lumberjack (HPC systems w/o outbound networking)
  - XD-SUBMIT (HPC systems with outbound networking)
- Third Phase:
  - HPC Annex (self-service HPC system tool)

Bringing Your Own Capacity to the Open Science Pool - Initial Experiments

Open Science Pool ("OSPool") Quick Overview (very simplified)

- HTCondor pool composed of federated compute resources from ~100 member sites
- Sites have full autonomy, donate spare resources from their own pools
- Powered by:
  - GlideinWMS (managed by OSG/PATh staff)
    - Submits "pilot jobs" – jobs that are Execution Points (EPs) that join the OSPool
    - Is triggered by user demand
  - HTCondor-CE (sometimes managed by OSG/PATh staff)
    - Accepts pilot jobs and runs them on the site's batch system

XD-SUBMIT

- Exploratory work: “what will it take to do this?"
- Hook up HPC centers to the Open Science Pool using the same infrastructure that we use for every other site
- Steps for OSG/PATh staff:
  - Obtain one user account from the HPC center for automated logins (SSH keypair, no 2FA)
  - Ask the user to add the new account to their allocation
  - Set up HTCondor-CE in front of the HPC site to accept pilot submissions and run them on the HPC site
  - Add GlideinWMS “factory” configuration so pilots with the correct parameters get submitted
  - Add GlideinWMS “frontend” configuration so the user's jobs can trigger pilot submission
  - Test!

Considerations for Allocation-Based Submission

- Only one project's jobs should result in pilot submission -- and only when desired
  - OSG/PATh staff need to keep track of user allocations and modify configs as needed
- Only jobs from that project should run on that pilot
  - Pilot needs to know what project "asked for it"
- Conversely: pilot should always be able to run jobs from that project
  - Idle pilots still consume allocations (SUs)
- Pilots should be shaped to fit the user's workflow
  - Affects queue choice, requested resources, environment (e.g., loaded modules)

Drawback: Poor User Isolation

- One account per site, all pilots use that account
- Users use separate allocations, but they all have the same file system permissions -- especially a problem with a shared file system
- Users can interfere with each other's files, or use up the account's disk quota

Drawback: Staff Effort Needed

- User has no control over the mechanisms for launching pilots
  - nor the number of pilots
  - nor the size of a pilot
  - nor the features
- User has no "panic button"
- User depends on OSG/PATh staff for changes
- Therefore OSG/PATh staff need to monitor closely to avoid wastage, and be quickly available to take care of user requests

Results and Lessons Learned

- ~1.5 million core-hours for Chemistry and Grav. Wave Astronomy jobs
- Learned how to run on several HPC sites
  - Each HPC site is unique -- queues, sbatch parameters, proxies, environment, filesystems, etc.
  - Tooling was written to adapt site environments to run our users' jobs (which has been reused elsewhere on the Open Science Pool)
- Experienced need for a "self-service" model
  - For proper isolation, users must be able to use their own identities, not just their allocations
  - Straightforward changes shouldn't require contacting staff

A Self-Service Tool

- Use what we learned about running pilots on these HPC systems, but start them as the researcher.
- Not hard to do interactively, so let's automate it.
- What about 2FA?
- Write a command-line tool that gives us the opportunity to prompt the researcher to log in.

HPC Annex

- Pilots are user-specific (and totally unshared).
- No staff involvement.
- Direct user control over allocation use and pilot “shape”; user can shut pilots off remotely.
- Pilots shut themselves off if idle.

- Requires users to (re-)authenticate each time they add their own capacity.

A Demonstration

› I'll mostly be following one of the recipes.

https://htcondor.com/web-preview/preview-ospool-byor/ospool/byor/stampede2

Status

- Deployed at the OSG Connect access points. ☺
  - Stampede 2, Bridges 2, Expanse, and Anvil supported.
- Deployed. ☹
  - Work is in progress to make enabling the tool turning a single knob, ideally one that's on by default.
- Targets a single use case.

Future Work

- Cover additional use-cases:
  - DAGMan
  - Sharing an annex with coworkers
- Eliminate external infrastructure and admin involvement.
  - Maybe no “home pool” required.
  - Could we offer user-specific access points as a service?

Questions?

Acknowledgements

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATh Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.