

Whither bps submit for Summit and Developers

DMLT vf2f

19 Oct 2022

Jim Chiang, Richard Dubois, Colin Slater

The question of which workflow system to back-end bps submit was [discussed](#) at the Data Facilities meeting on 2022-10-10. The outcome was to further discuss it at the following DMLT vf2f.

The charge for the DMLT discussion was:

- What are options for bps for the summit; and what should developers use

PanDA has been adopted for production processing, notably DRP, and is being used for routine HSC reprocessing at the USDF. Current status is that PanDA on s3df is not reliable enough yet for developers; nor is the monitoring easy to navigate without training.

Use of PanDA at the summit would also require a full PanDA server installation (in case of network outages); at present this is not in the cards.

Consequently, PanDA is not a candidate for either of these applications at present. Both of the usability issues need to be addressed, particularly if PanDA is to provide user batch.

This leaves HTCondor and Parsl to consider. It probably is the case that 2 options are a good idea to reduce risk if one option fails.

Assumptions:

- Prompt Processing will be the mechanism for FAFF's Rapid Monitoring
- Use at the summit is for exploration, needing quick turnaround for single users
- There is value in the summit solution being in use at the USDF for synergy reasons

Considerations

- HTCondor can submit to nodes without a separate workflow system; this is the mode used at NCSA and maybe the simplest
- Parsl can drive k8s nodes - no requirement for slurm either
- SLAC nodes will be under slurm (shared facility)
- Parsl is a single-user tool and uses pilot jobs. No sharing of pilot jobs. Also, cannot monitor other people's work (unless the path to the monitoring.db can be shared). Need good recommendations on usage to not waste nodes.
- HTCondor (without the need for glide-ins) would be "fire-and-forget", whereas the Parsl plugin (with an exception) continues to run Parsl at the user's command line.
- Parsl pilot jobs (and glide-ins) don't allow for the allocated resources to be dynamically sized as the compute load changes over the course of the pipeline.

What to do

- Find out whether HTCondor can share glide-ins at the USDF
 - Wei Yang will install a central HTCondor service; also need to improve the dev guide documentation
 - Need to better understand the interactions among bps, HTCondor and slurm (especially how it handles all the little jobs)
- Perform bake-off comparison of HTCondor and Parsl features
 - Should be clear on scope - are we also looking for multi-site capabilities, as backup to PanDA?