Stand Alone Project.py in AL9

Larsoft Coordination Meeting
Aug. 6, 2024

H. Greenlee

Outline

- Project.py overview / summary.
- How project.py was used historically in SL7.
- How MicroBooNE is using project.py currently using SL7 containers in AL9.
- UPS in AL9.
- Using project.py without containers in AL9.

Larbatch / Project.py Overview

- Larbatch is a ups product that provides a layer on top of jobsub_client / jobsub_lite.
  - Batch submission script project.py.
  - General purpose art / larsoft batch worker script condor_lar.sh.
  - Job configuration via xml file.
- Larbatch has historically been built as part of larsoft releases.
  - But larbatch is not a direct ups dependent of larsoft.
- Project.py has been used for production and analysis job submission by MicroBooNE (and still is).
  - Has been used by other experiments.
  - Other experiments are moving to fife_launch / fife_batch.

How Project.py was used Historically in SL7

- The larbatch ups product is experiment-agnostic and ups null-flavored.
- However using larbatch and project.py did and does require some experiment-specific configuration.
    - Project.py imports a python module called experiment_utilities.py containing experiment-specific configuraitons.
        - Still needed in AL9.
    - Project.py used to need to find the top level job fcl file to submit batch jobs.
        - This would normally be done via environment variable $FHICL_FILE_PATH by setting up the top level experiment ups product.
        - This requirement is removed in AL9.

How Project.py is used Currently by MicroBooNE

- MicroBooNE is currently using SL7 containers to submit batch jobs for both production and analysis, similar as when SL7 was the native OS.
  - Getting POMS to work with containers required some gymnastics.
  - MicroBooNE has worked around the fact that Fermilab doesn’t include jobsub_lite in the SL7 containers that it supplies by maintaining our own SL7 container that includes jobsub_lite.
  - This way of doing things is not ideal for various reasons.
    - High maintenance (need to keep rebuilding the MicroBooNE container).
    - May not be viable for long term (e.g. what if jobsub_lite team stops supplying SL7 rpms).

Stand Alone Project.py

- For reasons given on the previous slide, I have been investigating how to use project.py natively in AL9.
  - How to decouple larbatch from experiment software (as much as possible).
  - How to set up larbatch environmentally.
    - Execute path / python path.

UPS in AL9

- We have told that ups is not supported in AL9.
  - This is true, but there are caveats.
  - Ups scripts themselves, which are just bash scripts, still work.
  - The real problem is that there are no AL9-flavored ups products being built.
    - Art suite.
    - Nutools.
    - Larsoft.
  - It is still possible to initialize ups and set up null-flavored ups products natively in AL9.
    - sam_web_client
      - Works fine in AL9 as null-flavored ups product.
    - larbatch (project.py)
      - Works with recent updates.

Example – samweb

```
uboonegpvm03:~ $ cat /etc/redhat-release
AlmaLinux release 9.4 (Seafoam Ocelot)
uboonegpvm03:~ $ source /cvmfs/fermilab.opensciencegrid.org/products/common/etc/setup
uboonegpvm03:~ $ setup sam_web_client
uboonegpvm03:~ $ kx509    # Token authentication (htgettoken) also works
Authorizing ...... authorized
Fetching certificate ..... fetched
Storing certificate in /tmp/x509up_u1731
Your certificate is valid until: Wed Aug  7 16:26:25 2024
uboonegpvm03:~ $ samweb -s server-info
SAMWeb API for uboone
SAMWeb version: 3.1.0-pre1-32-g1e7ae68-dirty
Connected to: https://samweb.fnal.gov:8483/sam/uboone/api

Cherrypy version: 18.6.0
Backend server hostname: samweb-server-8c7dd799f-zm6ss
SQLAlchemy version: 1.3.19
Database: postgresql+psycopg2://samdbs:***@sampgsdb02.fnal.gov:5437/sam_uboone_prd
HTTP User-Agent: SAMWebClient/v3_3 (samweb.py) python/3.9.18

User information:
  Untrusted identity: greenlee@uboonegpvm03.fnal.gov
  Authenticated username: greenlee
  Roles: dataset, metadata, project
uboonegpvm03:~ $ samweb list-files --summary "defname: prod_bnb_swizzle_open_inclusive_v3"
File count:     24532
Total size:     25670846473143
Event count:    605230
```

Requirements for an AL9-Native Project.py

- Able to set up larbatch.
  - Can use ups. Larbatch is a null-flavored ups product.
- Access to samweb python api.
  - Can use ups. Sam_web_client is a null-flavored.
- Access to jobsub_lite.
  - Included in AL9 as Fermilab-supplied rpm.
- Able to run using system python (python 3.9 in AL9).
  - Yes, with tweaks.
    - Project.py has python 2/3 agnostic for long time.
- Access to experiment_utilities python module.
  - See following slides.
- Access to job fcl.
  - See following slides.

Accessing the Experiment_utilities Python Module in AL9

- In SL7, the experiment_utilities.py python module usually lived in some experiment-specific ups product.
  - Ups product ubutil in the case of MicroBooNE.
  - Ubutil is a flavored (non-null) ups product that can't be directly set up in AL9.
- One workaround is to manually set PYTHONPATH to include some place that has a copy of experiment_utilities.py.
  - Works, but not very convenient.
- To simplify things for MicroBooNE, I created a hand-maintained null-flavored ups product called ubutil_lite.
  - Includes latest copy of experiment_utilities.py.
  - Also has ups dependencies on larbatch and sam_web_client.

Accessing the Job Fcl

- Until recently, project.py needed to be able to find the job fcl(s) in the submit environment.
  - Normally done via $FHICL_FILE_PATH, which is set as a side effect of setting up some experiment-specific ups product.
    - Can't be done in AL9.
- Alternative ways of locating job fcl(s).
  - Set FHICL_FILE_PATH manually.
  - Specify absolute path of fcl in project.py xml file.
  - Specify <fclpath> in xml file.
- Not liking these workarounds, I modified project.py to not require finding the job fcl(s) in the submit environment.
  - Job will still run as long as fcl can be found in batch environment.
  - Project.py will still grab fcl from submit environment if it finds it.

Larbatch Status

- The latest version of larbatch is v01_60_00 (aka “current”).
  - Includes updates described on previous slide.
    - Compatible with AL9 system python 3.9.
    - Does not require access to job fcl when submitting jobs.

How to Use Project.py Natively in AL9 (Example)

```
uboonegpvm03:~ $ source /cvmfs/uboone.opensciencegrid.org/products/setup_uboone.sh
Setting LC_ALL=C
AlmaLinux 9.4 (Seafoam Ocelot)
uboonegpvm03:~ $ setup ubutil_lite
uboonegpvm03:~ $ ups active
Active ups products:
cetpkgsupport     v1_14_01        -f NULL                                    -z
/cvmfs/larsoft.opensciencegrid.org/products
jobsub_client     v_lite          -f NULL                                    -z
/cvmfs/fermilab.opensciencegrid.org/products/common/db
larbatch          v01_51_15       -f NULL                                    -z
/cvmfs/uboone.opensciencegrid.org/products
sam_web_client    v3_3            -f NULL                                    -z
/cvmfs/fermilab.opensciencegrid.org/products/common/db
ubutil_lite       v01_00_00       -f NULL                                    -z
/cvmfs/uboone.opensciencegrid.org/products
ups               v6_1_1          -f Linux64bit+3.10-2.17                    -z
/cvmfs/larsoft.opensciencegrid.org/products
uboonegpvm03:~ $ which project.py
/cvmfs/uboone.opensciencegrid.org/products/larbatch/v01_51_15/bin/project.py
uboonegpvm03:~ $ which samweb
/cvmfs/fermilab.opensciencegrid.org/products/common/prd/sam_web_client/v3_3/NULL/bin/samweb
```

What About Spack?

- There is currently no way to build or set up larbatch using spack (to my knowledge).
  - There is probably no fundamental reason why this couldn’t be done.
  - Evolving larbatch toward spack will probably need to wait until the code development problem has been better solved, and there is a regular spack distribution of larsoft.

Caveats and Limitations

- The batch worker script, condor_lar.sh, is still dependent on ups and SL7 containers.
  - Doesn't know anything about spack.
  - Updating condor_lar.sh to understand spack is within the realm of possibility, but no immediate plan to do so.
- Project.py is still tightly integrated with samweb.
  - Doesn't know anything about samweb successor products: rucio, metacat, data_dispatcher, etc. Unlikely to ever change.

Summary

- UPS not totally dead in AL9. Can still be used for setting up null-flavor products.
  - Including sam_web_client and larbatch.
- Latest version of larbatch works with AL9 system python (version 3.9).
- Partial decoupling of project.py from experiment-specific ups products.
  - Job fcl no longer required to submit jobs.
- I explained how to support experiment-specific python module experiment_utilities.py in AL9.