Interactive analysis on Bigmon

➢ What the user currently does for analysis:

1. Logon to lxplus
2. Checkout packages, e.g. AnalysisBase, from git
3. Customize local packages
4. Test custom code locally with a few input files
5. Submit a task
6. Monitor the task and kill/finish/retry it if necessary
7. Download the output
8. Test other custom code with downloaded files
9. Submit another task or run locally on all files to get results

➢ Seamless transition from 2-4 to 5-6 is ideal

- Task submission without custom code doesn’t really make sense for end-users

  • E.g. a user task

```bash
prun --cmtConfig=x86_64-centos7-gcc8-opt --nGBPerJob=10 --mergeOutput --useAthenaPackages --exec="runjob.sh
mc16_13TeV.364101.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV0_70_CFilterBVeto.derv.DAOD_HIGG4D3.e5271_s3126_r9364_p3978"
--inDS=mc16_13TeV.364101.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV0_70_CFilterBVeto.derv.DAOD_HIGG4D3.e5271_s3126_r9364_p3978
--outDS=ttt.huangruo_11_LegV02.mc16_13TeV.364101.Sh221-PDF30_Zmumu_MV0_70_CFBV.D3.e5271_s3122_r9364_p3978.smFin_L3
--outputs=hist:hist-output.root;HS:HS_common.root,txC:txC.root --writeInputToTxt=IN:input.txt --match="*" --mergeScript="xhadd %OUT %IN"
--inTarBall=jobcontents.tgz
```

  didn’t specify its analysis in --exec while including custom code in --inTarBall

  • pathena tasks are more like that except tasks with -trf which run official transformations out of the box
  • —trf provides a very similar function to prodsys where the user can construct analysis without any code-level customization by using options of the transformation like Reco_tf.py
Custom Code Dev

➢ Task submission will be triggered through a web browser running on a laptop
➢ Where the user develops the custom code
   - On the laptop
     • Easy for bigmon to make a tarball from local dir
     • The user has to setup analysis environment such as asetup, cvmfs, etc locally
       ➢ Could be a barrier for the user to use it
   - On a remote server
     • Analysis env is available at traditional facilities, e.g. lxplus@cern and acas@bnl
     • Bigmon has to create a tarball and upload using “ssh user@lxplus command” or reuse an existing tarball that was uploaded in a previous submission by pathena/prun
       ➢ Management of sensitive info
       ➢ Tokens with min permission?
Possible Task Submission Interface

Task/workflow management view with buttons

- New task
- Open task
- New workflow
- ...

To pop up interactive DAG drawer that helps the user to define a workflow visually and generate a workflow description file (more details in a future meeting)

Pop up

Task config view

Type: select
Host:
Work dir:
Athena ver:

Mandatory options: 
- inDS:
- outDS:
- outputs:
- exec:

Resource requirement options:
- nCore:
- memory:
- ...

Advanced options: +

Submit

Submission log view

Start
Making tarball ...
Done
Submitting the task ...
Done
TaskID=???

Link to the task summary view or real-time task logging view

146 options in pathena
135 options in prun

Click to show/hide relevant options

Internally triggers access to remote host
Real-time Task Logging View

➢ Current task summary view is useful for DPA or DevOps, but end-users don’t feel any real-time experiences since the view is rather static and there is nothing dynamic
  - Ultimately end-users are not interested in task attributes like prodSourceLabel and gshare. They want to see the system is working on their tasks and jobs are running/finishing
➢ JEDI and Panda server are producing messages when tasks and jobs change their status
➢ Possible to have a realtime logging view that consume messages related to a task?
  - Not polling, but push and consume
  - Old messages from ES and new messages via msg consumer in web browser
  - Visualization based on received messages, e.g. progress bars, continuous update of # of jobs in each status, ...