Using ntuplizer

Reynier Cruz Torres

March 25, 2021

Contents

1 Running jobs with the ntuplizer

Jobs are run on lxplus, then copied to and merged and stored on Cori. Lastly, the output files should be deleted from the grid. My lxplus and cori usernames are `rcruztor` and `reynier` respectively. Please, modify these for yourself accordingly. I colored them throughout the document so that you can quickly identify where to change the commands. Additionally, I am creating the examples below based on the job with identifier '19i3c2' and $\hat{p}_T$ 'pthat1'. Also update those specifically based on what you would like to run. Finally, this example uses the `VO_ALICE@AliPhysics::vAN-20201205_ROOT6-1` version of AliPhysics. This may be different for your job. You can find the specific version you need in the config file you will use.

1.1 Running jobs on lxplus

- ssh into lxplus: `ssh rcruztor@lxplus.cern.ch`

- clone the ntuplizer (only once): `git clone https://github.com/alwina/ntuple-gj.git`

- execute the following two commands (possibly add them to the bashrc file):

  - `source /cvmfs/alice.cern.ch/etc/login.sh`

  - `alienv enter VO_ALICE@AliPhysics::vAN-20201205_ROOT6-1`

- cd into the ntuplizer main directory

- run the ntuplizer `./macros/runNTGJ.C config/embed_19i3c2_pthat1.yaml full`

- check the status of your jobs by going to alimonitor.cern.ch, login, and go to the 'My jobs' tab

1.2 Copying, merging, and storing jobs on Cori

- `ssh -Y reynier@cori.nersc.gov`

- from lxplus (only once):

  `scp -r ~/.globus reynier@cori.nersc.gov:/global/homes/r/reynier/`

- add the following to your .bashrc file (only once):

```
#ALICE
if [ -d /cvmfs ];then
        export alien_CLOSE_SE="ALICE::LBL::EOS"
        source /cvmfs/alice.cern.ch/etc/login.sh
        module use /cvmfs/alice.cern.ch/x86_64-2.6-gnu-4.1.2/Modules/modulefiles
        alias alish="eval \$(alienv load AliPhysics)"
fi
```

- run the .bashrc file

- `alish` and if that does not work, do `module load AliPhysics` instead.

- `cd /project/projectdirs/alice/`reynier

- clone ntuplizer (only once): `https://github.com/alwina/ntuple-gj.git`

- cd into the ntuplizer main directory

- `alien-token-init` rcruztor

- To copy (on cori - note the 8 at the end of the second line):

  `./macros/alien_fastcopy \`

  `/alice/cern.ch/user/r/`rcruztor`/workingdir/outputdir_19i3c2_pthat1 \`

  `/global/project/projectdirs/alice/`reynier`/19i3c2/pthat1 8`

- To merge (on cori - note the `""` and `/*`):

  `./macros/MergeNtuple.C \`

  `"/global/project/projectdirs/alice/`reynier`/19i3c2/pthat1/*/*/AnalysisResults.root" \`

  `/global/project/projectdirs/alice/`reynier`/19i3c2/18q_embed_19i3c2_pthat1.root`

  * Replace /global/project/projectdirs/alice/reynier/19i3c2/pthat1 as you like

  * Be careful in the final merging step. You will overwrite any existing file of the same name. One option is to create the merged ntuple somewhere else and then move it into NTuples/embed when it's done.

1.3 Deleting copied files from the grid

From lxplus, depending on exactly which version of AliPhysics you have, you either do `aliensh` or `alien.py` (from anywhere). Then that'll take you into a shell from your grid home directory From there, you should be able to see workingdir, and inside there, the outputdir_XXX directories if you used aliensh, you have to use rmdir to remove a directory if you used alien.py, you just use rm -r like a normal shell from alimonitor.cern.ch, you can click "My home dir" in the top toolbar to navigate that from a web browser, but I have yet to figure out how to remove files through that. So instead we go through alien.py/aliensh

1.4   Checking job status after completion

From lxplus, depending on exactly which version of AliPhysics you have, you either do `aliensh` or `alien.py` (from anywhere). Then, type `masterjob <JobId>`. Here's the actual plethora of options:

```
usage: masterjob    <jobId> [-options] [merge|kill|resubmit|expunge]
options:
                -status <status>        :  display only the subjobs with that status
                -id <id>                :  display only the subjobs with that id
                -site <id>              :  display only the subjobs on that site
                -printid                :  print also the id of all the subjobs
                -printsite              :  split the number of jobs according to the execution site
                merge                   :  collect the output of all the subjobs that have already finished
                kill                    :  kill all the subjobs
                resubmit                :  resubmit all the subjobs selected
                expunge                 :  delete completely the subjobs
                You can combine kill and resubmit with '-status <status>' and '-id <id>'.
                For instance, if you do something like 'masterjob <jobId> -status ERROR_IB resubmit',
                 all the subjobs with status ERROR_IB will be resubmitted
```

1.5   Checking output files

- `root -l filename.root`

- `_file0->cd("AliAnalysisTaskNTGJ")`

- `_tree_event->GetEntries()`