

Table of Contents

Batch Jobs On Lxplus.....	1
Contents.....	1
Login to Lxplus.....	1
Batch Job Script.....	1
Job Submission.....	1
Checking Job Satus.....	2
Killing a Job.....	2
Submitting Multiple Batch Jobs.....	2
Links.....	3

Batch Jobs On Lxplus

Contents

- Login to Lxplus
- Batch Script
- Job Submission
- Checking Job Status
- Killing a Job
- Submitting Multiple Batch Jobs
- Links

Login to Lxplus

On lxplus you are limmited both in disk space and CPU time. In order to run medium to large size jobs from lxplus, you will need to submit a batch job using `bsub`. Your job will be submitted to a machine compatible with the machine you are submitting from, so for `CMSSW_1_3_X` and lower

```
ssh username@lxslc3.cern.ch
```

and for `CMSSW_1_4_X` and higher

```
ssh username@lxplus.cern.ch
```

Batch Job Script

Copy the following script into a file (say `lxplusbatchscript.csh`) and edit the file to run on your `.cfg` file. Also edit the last line to write into your CASTOR area (you have limited space on lxplus. If you don't use CASTOR you may lose your output!).

```
# Lxplus Batch Job Script
set CMSSW_PROJECT_SRC="cmssw_projects/13X/cmssw131hlt6/src"
set CFG_FILE="cfgs/steps2_3_4_5.cfg"
set OUTPUT_FILE="Analyzer_Output.root"
set TOP="$PWD"

cd /afs/cern.ch/user/s/ssimon/$CMSSW_PROJECT_SRC
eval `scramv1 runtime -csh`
cd $TOP
cmsRun /afs/cern.ch/user/s/ssimon/$CMSSW_PROJECT_SRC/$CFG_FILE
rfcp Analyzer_Output.root /castor/cern.ch/user/s/ssimon/$OUTPUT_FILE
```

Set the permissions on the script file with

```
chmod 744 lxplusbatchscript.csh
```

Job Submission

Now you can submit the job by using `bsub`, passing it the above script. An example command is

```
bsub -R "pool>30000" -q 1nw -J job1 < lxplusbatchscript.csh
```

There are a few arguments specified in this example

- `-R "pool>30000"` means you want a minimum free space of 30G to run your job.
- `-q 1nw` means you are submitting to the 1-week queue. Other available queues are:
 - ◆ 8nm (8 minutes)
 - ◆ 1nh (1 hour)
 - ◆ 8nh
 - ◆ 1nd (1day)
 - ◆ 2nd
 - ◆ 1nw (1 week)
 - ◆ 2nw
- `-J job1` sets job1 as your job name.
- `< lxplusbatchscript.csh` gives your script to the job.

See `man bsub` and the links for more info.

After entering the above command you will get the output

```
Job <557650> is submitted to queue <1nw>.
```

The unique job number 557650 is automatically generated.

Checking Job Satus

You can check the status of your job with the command `bjobs`. (To specify the job use `bjobs -J job1` or `bjobs 557650`)

```
bjobs
```

which gives the output

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
557650	ssimon	PEND	1nw	lxplus096		job1	Aug 9 16:16

You can see that this job is pending. To see more information about pending jobs use `bjobs -l`, and see `man bjobs` for more info.

Killing a Job

If you make a mistake and need to kill a job (i.e. submit to wrong queue), you can do so with

```
bkill jobnumber
```

To kill all your jobs:

```
bkill -u username 0
```

Submitting Multiple Batch Jobs

You can use `submitJobs.py.txt` script to submit your jobs (right click on it and save it, remove `.txt`). To use it:

1) Create one List.txt file with all the directories for all files you want to run on

For example for local files:

```
...
file:/afs/cern.ch/.../file_1.root
file:/afs/cern.ch/.../file_1.root
...
```

or for files in eos:

```
...
/store/group/.../file_1.root
/store/group/.../file_2.root
...
```

2) Place the script and the list in the same folder as your script you want it to use and give it permission (chmod 755)

3) Fill the customization area in the submitJobs.py file and run it. You do not need to make any directories, or empty them after the run.

submitJobs.py splits the List.txt file according to numbers you give it (it does not change the file itself). For example if you have 2111 lines in the List.txt file, you can tell it to do 11 jobs per 200 files -- the 11th job will run on 111 files. You do not need to create any directories, it creates tmp directory which contains smaller txt lists and job.sh scripts that are sent with bsub.

After the jobs are done you can see the logs files in tmp folder. This directory is cleared at the start of submitJobs.py script. The result files are created in res directory, which is not cleared. Remember to change the base of the output file name when you submit new jobs. The output files are named according to the string you give in the submitJobs.py script, if you give "outputFile", you will get outputFile_1.root, outputFile_2.root and so on.

Links

- [LSF Quick Intro](#)
- [Helpful Twiki](#)
- [lxbatch page](#)

-- DavidCockerill - 27-Apr-2011

This topic: Main > BatchJobs

Topic revision: r6 - 2016-09-20 - AndreyKupich



Copyright &© 2008-2024 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback