

# Running LArSoft Grid jobs at CERN

Christoph Alt

January 31<sup>st</sup>, 2018 / DUNE collaboration meeting @ CERN

**ETH** zürich

WA105 

**DUNE** DEEP UNDERGROUND  
NEUTRINO EXPERIMENT

## What is...

- the grid?
- the CERN Tier 0 (T0)?
- HTCondor?

# The Worldwide LHC Computing Grid (WLCG)

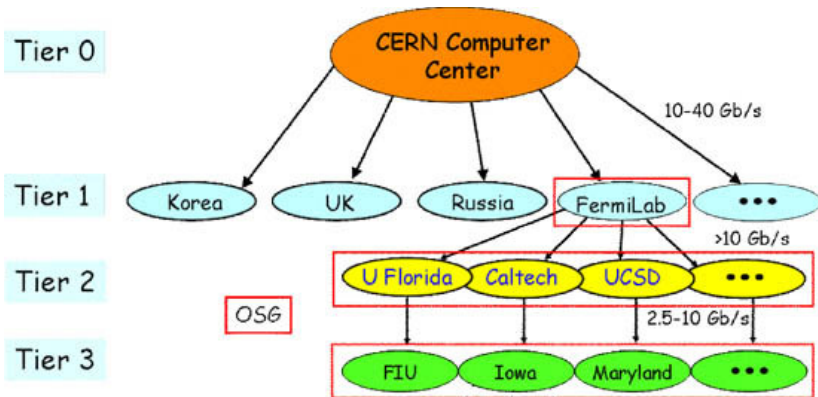


Figure : Paul Avery

- CERN T0 is a synonym for the CERN Data Centre
  - located at CERN and Wigner Research Centre for Physics in Budapest
  - the two sites are connected by two dedicated 100 Gbit/s data links
- CERN T0 was developed for LHC experiments
  - ~ 7.500 servers, 115.00 cores and 45.000 disks
- everyone with a CERN account can use part of T0 resources
  - both NP02 and NP04 have dedicated cores and disk space from CERN T0: 1 PB EOS, 6 PB tape and 1500 cores
  - to use them, select np-comp as primary group and subscribe to np02-t0comp-users or np04-t0comp-users e-group

# Batch system: HTCondor

You can submit jobs to the CERN T0 through HTCondor:

1. a user submits a job
2. HTCondor adds the job to the job queue
3. the job is executed on a worker node of the CERN T0

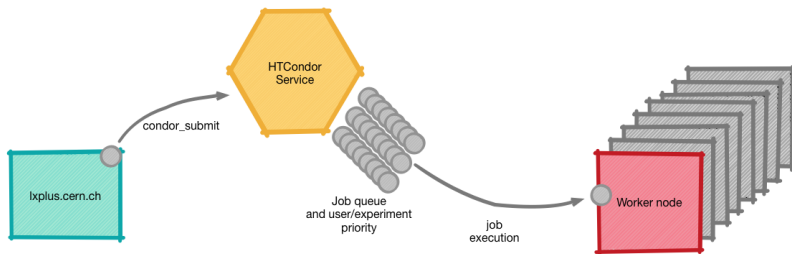


Figure : CERN

# Batch system: data flow

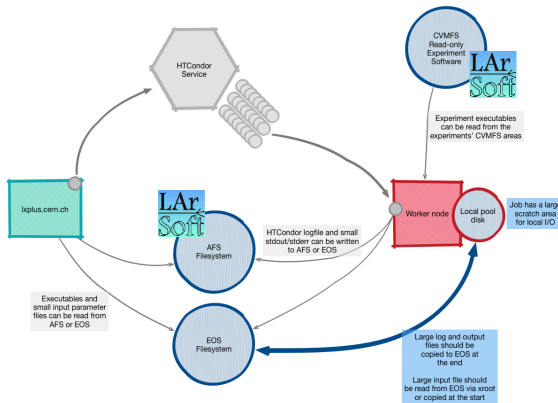


Figure : CERN (edited)

- LArSoft: weekly release on CMVFS or local install. on AFS
- Input and output files: AFS or EOS (large files: EOS)

## When to use the CERN T0

- Basically always when you want to run many jobs in parallel
- Example: data reconstruction
- offline: with the 3x1x1, we have recorded 1718 subruns (335 events per subrun) and it takes  $\sim 2$  hours to reconstruct one subrun in LArSoft on a lxplus machine  $\rightarrow 143$  days
- other examples: light map generation, large MC sample generation

# How to run a LArSoft job on the CERN T0

The following scripts are based on Nektarios Benekos' (Neutrino Platform, CERN) example scripts: <https://gitlab.cern.ch/protoDune/NP-Tier0-ProductionScripts>

You need three things:

1. A (bash) script running the LArSoft command: "Example1.sh"
2. A job submission script that refers to Example1.sh:  
"Example1.sub"
3. A text file holding input arguments for the bash script (e.g. input file)



# Example 1: using a weekly dune/tpc release on CVMFS

## 1: bash script that runs LArSoft

(" /afs/cern.ch/work/c/chalt/public/CERNTutorial/Example1.sh")

```
1 #!/bin/bash
2
3 # define and setup weekly release of dune/tpc
4 source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
5 version=v06_64_00
6 setup dune/tpc ${version} -q e14:prof
7 export EOS_MGM_URL=root://eospublic.cern.ch
8
9 # get input
10 export InputFile=$1
11
12 # isolate run number and subrun number from input file
13 echo "Input file: "$InputFile
14 tmp=${InputFile##*/}
15 RunNumber=${tmp%*/}
16 tmp=${InputFile##*/}
17 SubrunNumber=${tmp%.root}
18 echo "Run number: "$RunNumber
19 echo "Subrun number: "$SubrunNumber
20
21 # define and create output path
22 export EOSOutputPath=/eos/user/c/chalt/CERNTutorialOutput/$RunNumber
23 mkdir -p $EOSOutputPath
24
25 # define output file
26 export EOSOutputFile=$EOSOutputPath/$RunNumber-$SubrunNumber.root
27
28 # name of the FHiCL that you want to run
29 export FHiCL=rawhitfinding_reco_3x1x1dp.fcl
30
31 # run LArSoft
32 lar -c $FHiCL $InputFile -n 1 -o $RunNumber-$SubrunNumber-Reco.root # this will create the output file on the scratch disk.
33 # After job completion, the output file is then copied to eos.
34
35 # copy output file to eos and delete other output files
36 scp $RunNumber-$SubrunNumber-Reco.root $EOSOutputPath
37 rm -f $RunNumber-$SubrunNumber-Reco.root
38 rm -f debug.log
39 rm -f hist.root
```

run it: source Example1.sh /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-0.root

## 2: submission script

(" /afs/cern.ch/work/c/chalt/public/CERNTutorial/Example1.sub")

```
1 notify_user          = christoph.alt@cern.ch
2 ## Never, Error, Complete, Always
3 notification         = Always
4 ##-----##
5 ## universe means "job type"
6 ## The universe defines an execution environment. You will almost always use VANILLA.
7 ## http://research.cs.wisc.edu/htcondor/manual/v7.8/2_4Road_map_Running.html#SECTION00341200000000000000
8 universe             = vanilla
9 ##-----##
10 ## path to program to start. Could be relative
11 executable           = /afs/cern.ch/work/c/chalt/public/CERNTutorial/Example1.sh
12 ##-----##
13 ## command line args
14 #arguments           = input
15 ##-----##
16 JOBNAME              = Example1
17 output               = $(JOBNAME).$(ClusterId).$(ProcId).out
18 error                = $(JOBNAME).$(ClusterId).$(ProcId).error
19 log                  = $(JOBNAME).$(ClusterId).$(ProcId).log
20 getenv               = True
21 #####
22 # check with 'condor_version':
23 #If the release is 8.5.8 or greater, you don't need send_credential
24 # in the near future most condor clients shouldn't need this option, so it will be obsolete
25 #send_credential     = True
```

```

26 #####
27 ##-----##
28 ## Force HTCondor not to use shared filesystem
29 should_transfer_files = YES
30 ##-----##
31 when to transfer_output = ON_EXIT
32 initialdir = /afs/cern.ch/work/c/chalt/public/CERNTutorial/Output/
33 ##transfer_input_files = /afs/cern.ch/work/c/chalt/public/LArSoftWA105off/srcs/dunetpc/fcl/3x1x1dp/reco/rawhitfinding_reco_3x1x1dp.fcl
34 requirements = (Arch == "INTEL" && OpSys == "LINUX") || (Arch == "X86_64" && OpSys == "LINUX" )
35 #|| (OpSysAndVer =?= "SLCern6" ) || ( (OpSysAndVer =?= "CentOS7") && (CERNEnvironment =?= "qa") )
36 priority = 20
37 ## By default, a job will get one CPU, 2 GB memory and 20 GB disk space. There is a limit of 2 GB memory / CPU.
38 ##-----##
39 ## number of cores to reserve
40 #request_cpus = 8
41 ##-----##
42 ## scratch disk to reserve
43 #request_disk = 10 GB
44 ##-----##
45 ## if your program needs a certain amount of total RAM memory or free disk space, you can use these commands to force that your
46 ## jobs will be only executed on machines with at least the requested memory/free disk space
47 ## For example, if your program needs at least 1.5 GB of RAM and 5 GB of free space in disk, and due to library dependencies it
48 ## can only run on machines with Linux Centos7 or above, add next commands in your submit file:
49 ## request_memory = 1.5 GB
50 ## requirements = (OpSys == "Linux") && (OpSysAndVer == "CentOS7")
51 ##-----##
52 ## HTCondor queues
53 ## espresso = 20 minutes, microcentury = 1 hour, longlunch = 2 hours, workday = 8 hours, tomorrow = 1 day, testmatch = 3 days, nextweek = 1 week
54 +JobFlavour = "espresso"
55 ## Instead of JobFlavour, you can also specify "MaxRuntime" (in seconds) for short jobs
56 ##+MaxRuntime = 1200
57 queue arguments from RunList.txt

```

### 3: RunList.txt:

```

1 /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-0.root
2 /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-1.root
3 /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-2.root

```

- run it: `condor_submit Example1.sub`

- This will submit as many jobs as there are lines in RunList.txt.  
In our case:

- Example1.sh /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-0.root
- Example1.sh /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-1.root
- Example1.sh /eos/experiment/wa105/offline/LArSoft/Data/Raw/840/840-2.root

```
[chalt@lxplus040 CERNTutorial]$ condor_submit Example1.sub
Submitting job(s)...
3 job(s) submitted to cluster 111731.
[chalt@lxplus040 CERNTutorial]$ condor_q

-- Schedd: bigbird05.cern.ch : <128.142.165.14:9618?... @ 01/22/18 17:35:09
OWNER BATCH_NAME          SUBMITTED   DONE    RUN    IDLE  TOTAL JOB_IDS
chalt CMD: Example1.sh    1/22 17:35      -     -     3      3 111731.0-2

3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended
[chalt@lxplus040 CERNTutorial]$
```

After job completion, the three reco output files should appear on eos:

```
[chalt@lxplus040 CERNTutorial]$ ls -lh /eos/user/c/chalt/CERNTutorialOutput/840/
total 15M
-rw-r--r--. 1 chalt np-comp 2.7M 22. Jan 17:39 840-0-Reco.root
-rw-r--r--. 1 chalt np-comp 2.7M 22. Jan 17:39 840-1-Reco.root
-rw-r--r--. 1 chalt np-comp 2.1M 22. Jan 17:46 840-2-Reco.root
[chalt@lxplus040 CERNTutorial]$
```

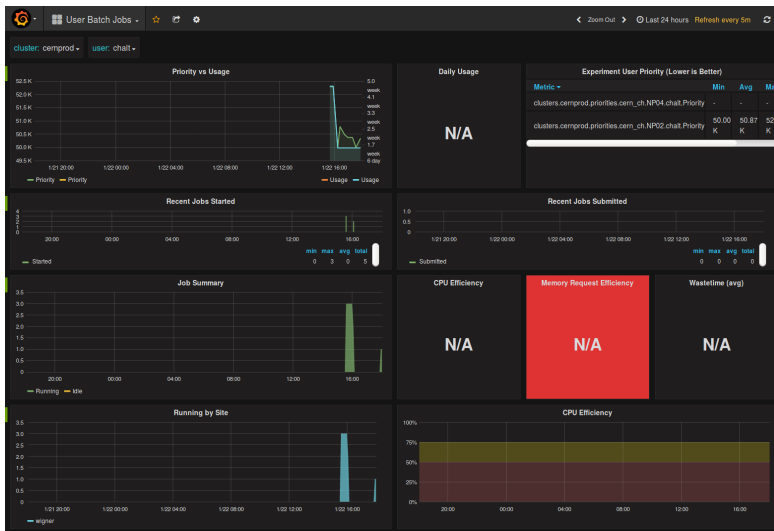
## Example 2: using a local LArSoft installation (afs or neut)

- See Example2.sh and Example2.sub in  
/afs/cern.ch/work/c/chalt/public/CERNTutorial/

```
1 #!/bin/bash
2
3 # define and setup larsoft installation
4 export myworkpath=/afs/cern.ch/work/c/chalt/public/LArSoftWA105off
5 version=v06_63_00
6 source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
7 source $myworkpath/localProducts_larsoft_${version}_e14_prof/setup
8 mrbslp
9 export EOS_MGM_URL=root://eospublic.cern.ch
```

# Monitoring your grid jobs

<https://batch-carbon.cern.ch/grafana/dashboard/db/user-batch-jobs?var-cluster=cernprod&var-user=chalt>



## Useful links:

- CERN Neutrino Platform TWiki: <https://twiki.cern.ch/twiki/bin/view/CENF/NeutrinoPlatformCAF>
- CERN Batch Service User Guide:  
<http://batchdocs.web.cern.ch/batchdocs/index.html>
- HTCondor User's Manual: [http://research.cs.wisc.edu/htcondor/manual/v7.6/2\\_Users\\_Manual.html](http://research.cs.wisc.edu/htcondor/manual/v7.6/2_Users_Manual.html)
- Monitoring experiment resources:  
<https://batch-carbon.cern.ch/grafana/dashboard/db/experiment-batch-details?var-cluster=cernprod&var-experiment=NP02>