Submitting Jobs Tutorial

Madison Bootcamp, June 7th 2017

G. Merino, D. Schultz

before starting…

let us prepare our digital certificates

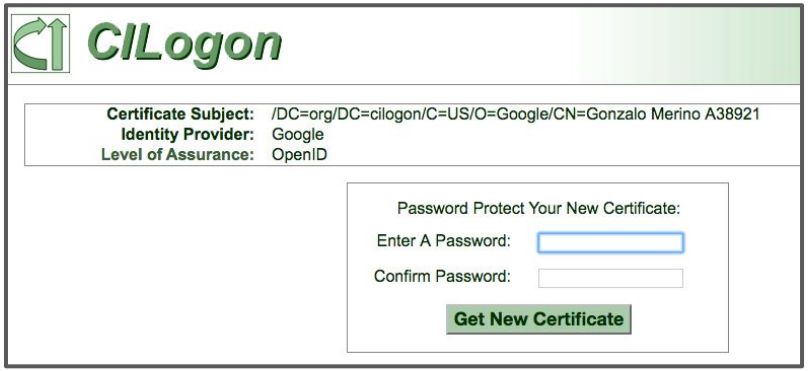
for remote file access

File access: GridFTP & digital certificates

To access files in Madison remotely you will have to use GridFTP ⇒ Need a digital certificate for authentication/authorization.

If you do not have already one, please go to: *https://cilogon.org/*

- You can search for your home institution IdProvider … or just use Google.



Copy the generated **`usercred.p12`** certificate file to your WIPAC home dir:

```
$ scp Downloads/usercred.p12
pub.icecube.wisc.edu:.globus/usercred.p12
```

File access: GridFTP & digital certificates

```
sub-1 ~ $ openssl pkcs12 -in .globus/usercred.p12 -nokeys | grep subject
Enter Import Password:
MAC verified OK
subject=/DC=org/DC=cilogon/C=US/O=Google/CN=Gonzalo Merino A38921
```

Copy-paste this certificate subject to your LDAP profile (the string in red above)

https://internal.icecube.wisc.edu/directory/

Intro to Condor

Condor clusters

IceCube has an HTCondor compute cluster at UW-Madison, named NPX:

- ~7600 HT CPU cores, ~400 GPUs
- Use **submitter.icecube.wisc.edu** to submit your jobs to NPX

```
mylaptop ~ $ ssh pub.icecube.wisc.edu

pub1 ~ $ ssh submitter.icecube.wisc.edu
```

submitter.icecube.wisc.edu

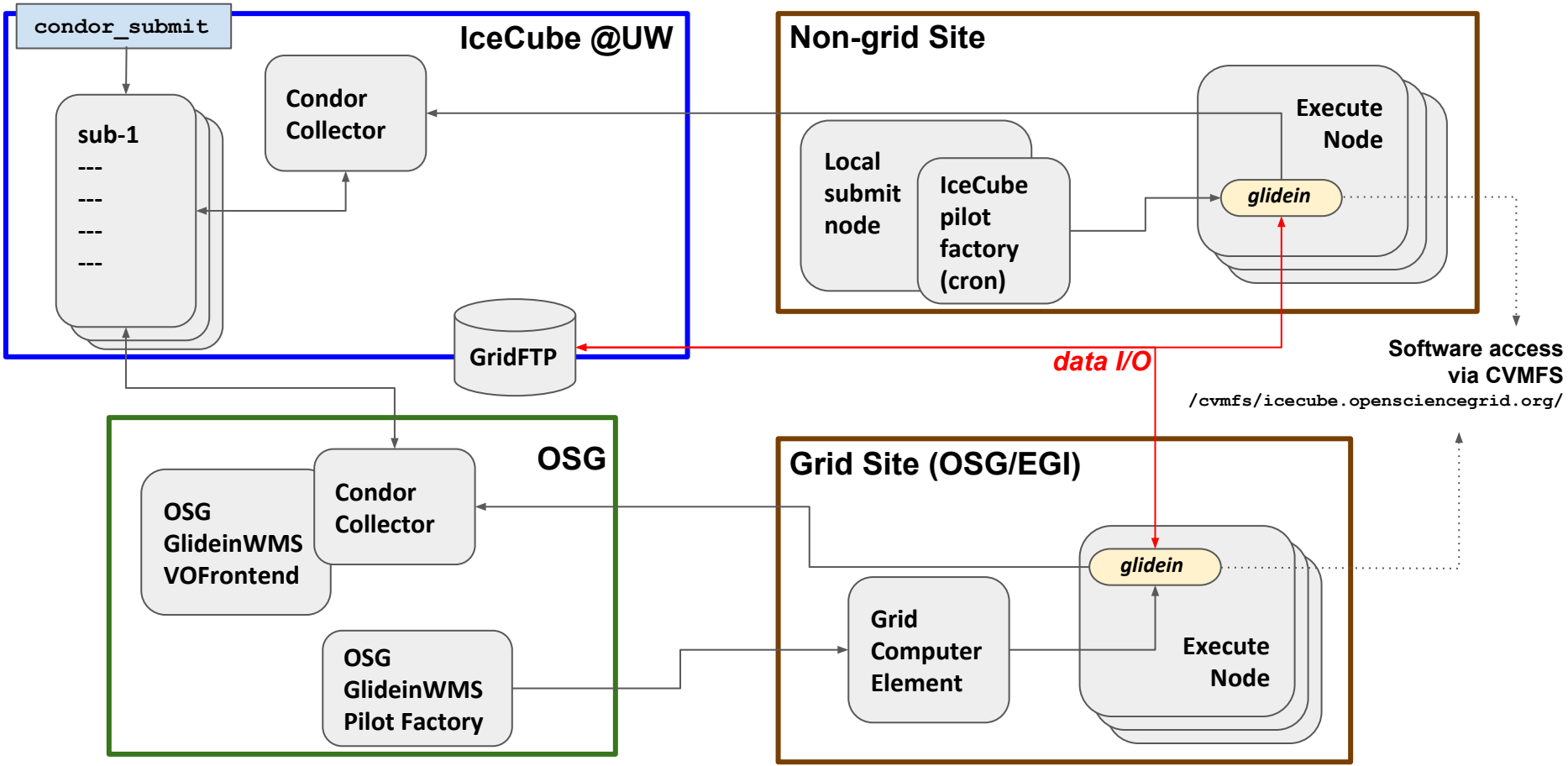
```
submitter ~ $ pwd
/home/gmerino


submitter ~ $ ls /data/
ana  exp  sim  user  wipac


submitter ~ $ ls /cvmfs/icecube.opensciencegrid.org/
buildall.sh  data  distrib  iceprod  py2-v1  py2-v2  py2-v3
py2-v3_early_access  README  setup.sh  standard


submitter ~ $ mkdir /scratch/gmerino
```

IceCube Grid



Grid submit host: sub-1

Use **sub-1.icecube.wisc.edu** to submit Condor jobs to the Grid.

Should look pretty much the same as submitter, except:

- No /data/sim, ana, exp, user directories mounted

Basic idea: The nodes out there where your Grid jobs will run will not have direct access to /data/exp, etc. The setup at sub-1 tries to mimic that.

Points to keep in mind when running jobs in the Grid:

- Do not assume direct access to files in /data/exp,sim,ana,user
- Try to keep your resources small, you will get more CPUs (RAM<2GB, disk<30GB, job duration 1 to few hours)

Simple job script

Advice: Job logfiles in network filesystems can generate instability in HTCondor.

Please keep your job logfiles in local disk.

```
sub-1 ~ $ mkdir /scratch/gmerino/; cd /scratch/gmerino/
sub-1 ~ $ wget  http://icecube.wisc.edu/~gmerino/bootcamp/job.sh
```

```
sub-1 ~ $ cat job.sh
#!/bin/bash
printf "Start time: "; /bin/date
printf "Job is running on node: "; /bin/hostname
printf "Job running as user: "; /usr/bin/id
printf "Job is running in directory: "; /bin/pwd
echo "Working hard..."
sleep $1
echo "Job complete!"
```

```
sub-1 ~ $ chmod +x job.sh
sub-1 ~ $ ./job.sh 5
```

Submit File

```
sub-1 ~ $ cat job.sub
executable = job.sh
arguments = 10

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_memory = 100MB
request_disk = 1GB
#request_gpus = 1

queue 1
```

List your executable and any arguments it takes.

Log: file created by HTCondor to track job progress

output/error: captures stdout/stderr

Request the appropriate resources for your job to run.

queue: keyword indicating “create a job”

Resource Request

If you do not specify resource requirements (cpus, memory, disk) the job will grab the default for the site where it will end up running.

Most sites will give you 1 cpu, ~2GB RAM, ~20GB disk… but this can vary a lot.

Some sites can have smaller defaults! → your jobs will be killed if they overuse resources.

It is very important to request appropriate resources for a job

- Too little → your job might be killed for overusing resources
- Too much → your job will wait forever. There will be fewer “job slots” matching your requirements.

Run test jobs and use the log file to request the right amount of resources.

Always good to check for CVMFS

```
sub-1 ~ $ cat job.sub
executable = job.sh
arguments = 10

log = job.log
output = job.out
error = job.err

request_cpus = 1
request_memory = 100MB
request_disk = 1GB

Requirements = HAS_CVMFS_icecube_opensciencegrid_org

queue 1
```

Submitting and monitoring

To submit and check for jobs: condor_submit and condor_q

```
sub-1 ~ $ condor_submit job.sub
Submitting job(s).
1 job(s) submitted to cluster 12898721.
```

```
sub-1 ~ $ condor_q
-- Schedd: sub-1.icecube.wisc.edu : <128.104.255.232:58276?...
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
12898721.0   gmerino     6/13 19:32   0+01:45:39 R  0   1464.8 job.sh 100
```

http://research.cs.wisc.edu/htcondor/manual/v8.7/condor_submit.html

http://research.cs.wisc.edu/htcondor/manual/v8.7/condor_q.html

Submitting and monitoring

```
sub-1 ~ $ condor_q
-- Schedd: sub-1.icecube.wisc.edu : <128.104.255.232:58276?...
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
12898721.0   gmerino     6/13 19:32   0+01:45:39 R  0   1464.8 job.sh 100
```

A Job **Cluster** identifies a batch of jobs.

Individual jobs within a Cluster are identified by the number after the dot: the

NOTE: Use `condor_q -allusers` if you want to see other users jobs in the queue.

Submitting and monitoring

```
sub-1 ~ $ condor_q
-- Schedd: sub-1.icecube.wisc.edu : <128.104.255.232:58276?...
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
12898721.0   gmerino      6/13 19:32   0+01:45:39 R  0   1464.8 job.sh 100
```

Most common job status cycle:

Idle (I): Job has not started yet… waiting in the queue.

→ **Running** (R): Job is running...

→ **Completed**: If the job has completed, it will not appear in condor_q

→ **Held** (H): Stalled jobs. Something YOU need to fix.

A job that goes on hold is interrupted (all progress is lost) and kept from running again. It remains in the queue in the “H” state.

Held jobs removed e-mail

```
From: root <root@sub-1.icecube.wisc.edu>
Date: 14 June 2016 at 04:17
Subject: [htcondor] sub-1: held jobs removed
To:

12925663.0   gmerino         6/13 23:54 Policy violation. Memory limit exceeded: 2004 MB resident > 2000
MB requested. (by user condor)
12925674.0   gmerino         6/13 23:54 Policy violation. Memory limit exceeded: 2003 MB resident > 2000
MB requested. (by user condor)
12925692.0   gmerino         6/13 23:54 Policy violation. Memory limit exceeded: 2005 MB resident > 2000
MB requested. (by user condor)
```

We periodically scan for held jobs in the queue, remove them and notify users via email.

Some other typical hold reasons ...

```
12276425.0   gmerino      6/3  00:05 Error from slot1@e201.chtc.wisc.edu: Job failed to complete in 72 hrs

12015071.5   gmerino         3/23 14:33 Error from glidein_7164_75405897@a0437: STARTER at 10.80.2.181
failed to send file(s) to <128.104.255.232:59904>: error reading from
/home/icecu038/home_cream_966794713/CREAM966794713/glide_aXZBCc/execute/dir_34476/_condor_stdout: (errno
2) No such file or directory; SHADOW failed to receive file(s) from <134.93.174.12:35154>
```

Log File

```
sub-1 ~ $ cat job.log
000 (12898721.000.000) 06/13 21:50:14 Job submitted from host:
<128.104.255.232:58276?addrs=128.104.255.232-58276>
...
001 (12898721.000.000) 06/13 21:52:33 Job executing on host:
<144.92.166.137:27680?addrs=144.92.166.137-27680>
...
006 (12898721.000.000) 06/13 21:52:33 Image size of job updated: 1
      0  -  MemoryUsage of job (MB)
      0  -  ResidentSetSize of job (KB)
...
005 (12898721.000.000) 06/13 21:52:43 Job terminated.
      (1) Normal termination (return value 0)
            Usr 0 00:00:00, Sys 0 00:00:00  -  Run Remote Usage
            Usr 0 00:00:00, Sys 0 00:00:00  -  Run Local Usage
            Usr 0 00:00:00, Sys 0 00:00:00  -  Total Remote Usage
            Usr 0 00:00:00, Sys 0 00:00:00  -  Total Local Usage
      314  -  Run Bytes Sent By Job
      281  -  Run Bytes Received By Job
      314  -  Total Bytes Sent By Job
      281  -  Total Bytes Received By Job
      Partitionable Resources :    Usage  Request Allocated
         Cpus                 :                1         1
         Disk (KB)            :       12   102400    940361
         Memory (MB)          :        0      100       100
```

Output File

```
sub-1 ~ $ cat job.out
Start time: Wed Jun  7 15:53:27 GMT 2017
Job is running on node: hibat0106.cmsaf.mit.edu
Job running as user: uid=10125(osg01) gid=10125(osg01) groups=10125(osg01),10005(osg)
Job is running in directory:
/export/data1/condor/execute/dir_144460/glide_Rvbozm/execute/dir_111261/glidein/execute.18.12.6.106-112180/di
r_122685

Working hard...
Job complete!
```

Finding Job Attributes

```
sub-1 ~ $ condor_q -long 12887688.0
JobStatus = 2
LastJobStatus = 1
User = "gmerino@icecube.wisc.edu"
Err = "/scratch/gmerino/Data/2013/logs/12887688.err"
Out = "/scratch/gmerino/Data/2013/logs/12887688.log"
NumJobStarts = 1
Args = "-g
/data/exp/IceCube/2013/filtered/level2/0505/Run00122300/Level2_IC86.2013_data_Run00
122300_0505_0_9_GCD.i3.gz -i
/data/exp/IceCube/2013/filtered/level2/0505/Run00122300/Level2_IC86.2013_data_Run00
122300_Subrun00000070.i3.bz2 -o
/data/ana/Cscd/StartingEvents/exp/IC86_2013/burnsample/l3/00122300/Level2_IC86.2013
_data_Run00122300_Part00000070.i3.bz2"
RemoteHost = "slot1@e281.chtc.wisc.edu"
ResidentSetSize_RAW = 1200308
DiskUsage_RAW = 891690
RemoteUserCpu = 7669.0
```

Useful Job Attributes

`JobStatus`: number indicating Idle (1), Running (2), Held (5), etc

`RemoteHost`: where the job is running

`ResidentSetSize_RAW`: Maximum observed physical memory in use by the job in KiB while running.

`DiskUsage_RAW`: Maximum observed physical memory in use by the job in KiB while running.

`RemoteUserCpu`: The total number of seconds of user CPU time the job has used.

`EnteredCurrentStatus`: time of last status change

`NumJobStarts`: number of times the job started executing

http://research.cs.wisc.edu/htcondor/manual/v8.5/12_Appendix_A.html

Displaying Job Attributes

Use the "`-autoformat`" option for condor_q

```
sub-1 $ condor_q -af JobStatus ClusterID ProcId RemoteHost ResidentSetSize_RAW

2 12892531 0 slot1@glidein_235900_283164684@cabinet-0-0-7.t2.ucsd.edu 1272208
2 12892986 0 glidein_10345_623188368@jux7c.zeuthen.desy.de 1290364
2 12893002 0 slot1@e137.chtc.wisc.edu 1181296
```

The "`-constraint`" option can also be handy

```
sub-1 $ condor_q -c jobstatus==2 -af JobStatus ClusterID ProcId RemoteHost
ResidentSetSize_RAW

2 12892531 0 slot1@glidein_235900_283164684@cabinet-0-0-7.t2.ucsd.edu 1272208
2 12892986 0 glidein_10345_623188368@jux7c.zeuthen.desy.de 1290364
2 12893002 0 slot1@e137.chtc.wisc.edu 1181296
```

DAGMan

DAGMan is a tool that comes bundled with HTCondor. It can do two useful things:

- Control number of running jobs
- Handle inter-job dependencies

DAGMan Basics

A basic DAG submit file:

```
submitter ~ $ cat dagman.dag
JOB job1 job.sub
VARS job1 Seconds="100"
JOB job2 job.sub
VARS job2 Seconds="200"
JOB job3 job.sub
VARS job3 Seconds="15"
JOB job4 job.sub
VARS job4 Seconds="20"
```

Modify a bit your demo job.sub submit file

```
sub-1 ~ $ cat job.sub
executable = job.sh
arguments = $(Seconds)

log = job.log
output = job.$(Cluster).out
error = job.$(Cluster).err

request_cpus = 1
request_memory = 100MB
request_disk = 1GB

Requirements = HAS_CVMFS_icecube_opensciencegrid_org

queue
```

DAGMan Basics

Submit your DAG, limiting to 2 active jobs:

```
sub-1 ~ $ condor_submit_dag -maxjobs 2 dagman.dag

------------------------------------------------------------------------
File for submitting this DAG to Condor           : dagman.dag.condor.sub
Log of DAGMan debugging messages                 : dagman.dag.dagman.out
Log of Condor library output                     : dagman.dag.lib.out
Log of Condor library error messages             : dagman.dag.lib.err
Log of the life of condor_dagman itself          : dagman.dag.dagman.log

Submitting job(s).
1 job(s) submitted to cluster 21135967.
------------------------------------------------------------------------
```

DAGMan Basics

Looking at the dagman output messages, the job limit can be seen:

```
...
06/02/14 14:40:49 Of 4 nodes total:
06/02/14 14:40:49  Done   Pre   Queued Post   Ready   Un-Ready   Failed
06/02/14 14:40:49   ===   ===      === ===  ===      ===      ===
06/02/14 14:40:49     0    0        2    0    2        0        0
06/02/14 14:40:49 0 job proc(s) currently held
06/02/14 14:40:49 Note: 2 total job deferrals because of -MaxJobs limit (2)
...
```

DAGMan Basics

It can be useful to have this "**maxjobs**" parameter as a config var of the DAG:

```
sub-1 ~ $ cat dagman.dag
JOB job1 job.sub
VARS job1 Seconds="100"
JOB job2 job.sub
VARS job2 Seconds="200"
JOB job3 job.sub
VARS job3 Seconds="15"
JOB job4 job.sub
VARS job4 Seconds="20"
CONFIG dagman.config

sub-1 ~ $ cat dagman.config
DAGMAN_MAX_JOBS_SUBMITTED = 2
```

DAGMan Basics

Then you will be able to modify it while the DAG is running: The DAG will re-read its config file upon hold/release.

```
submitter ~ $ condor_hold 21135967


submitter ~ $ vi dagman.config


submitter ~ $ cat dagman.config
DAGMAN_MAX_JOBS_SUBMITTED = 4


submitter ~ $ condor_release 21135967
```

Job Dependencies

Now let us look at an example with dependencies, where one job must run before another one.

Let's make a DAG with 3 parents and one child (maybe processing and cleanup?):

```
submitter ~ $ cat dagman.dag
JOB job1 job.sub
VARS job1 Seconds="100"
JOB job2 job.sub
VARS job2 Seconds="200"
JOB job3 job.sub
VARS job3 Seconds="15"
JOB job4 job.sub
VARS job4 Seconds="20"
# define the DAG relationship
Parent job1 job2 job3 Child job4
```

Job Dependencies

```
submitter ~ $ condor_submit_dag -maxjobs 2 dagman.dag
```

Looking at the dagman output messages, we can see the child job is un-ready until the three parents have finished:

```
...
06/02/14 15:45:31 Of 4 nodes total:
06/02/14 15:45:31  Done   Pre   Queued    Post   Ready   Un-Ready   Failed
06/02/14 15:45:31   ===   ===       ===   ===   ===       ===       ===
06/02/14 15:45:31    0     0         2     0     1         1         0
06/02/14 15:45:31 0 job proc(s) currently held
06/02/14 15:45:31 Note: 1 total job deferrals because of -MaxJobs limit (2)
...
06/02/14 15:45:56 Of 4 nodes total:
06/02/14 15:45:56  Done   Pre   Queued    Post   Ready   Un-Ready   Failed
06/02/14 15:45:56   ===   ===       ===   ===   ===       ===       ===
06/02/14 15:45:56    3     0         0     0     1         0         0
06/02/14 15:45:56 0 job proc(s) currently held
...
```

An IceTray Grid Job

IceTray remote file staging

IceTray can work with URLs as well as with local paths, and everything just works

```
gsiftp://gridftp.icecube.wisc.edu/data/exp/IceCube/2015/.../somefile.i3.bz2
```

Reading from remote files: File stager will issue the "**gridftp** download" command to stage the file into a temporary local directory, and provide a file handle to icetray.

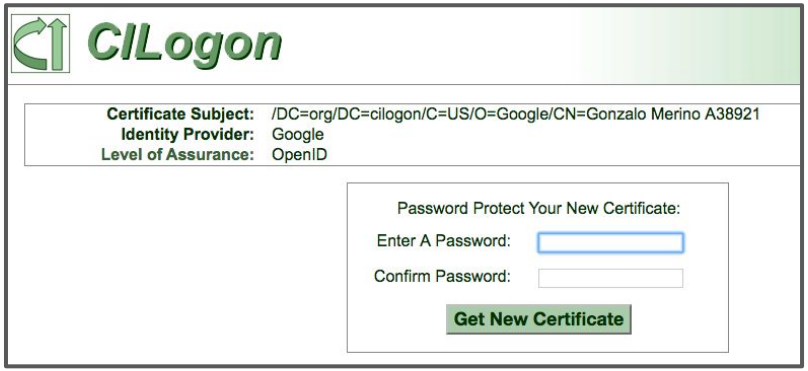
Writing to remote files: Icetray will write the output into a temporary local directory, and File stager will issue a "**gridftp** upload" once the file handle is released.

File access: GridFTP & digital certificates

To access files in Madison remotely you will have to use GridFTP ⇒ Need a digital certificate for authentication/authorization.

If you do not have already one, please go to: *https://cilogon.org/*

- You can search for your home institution IdProvider … or just use Google.



Copy the generated **`usercred.p12`** certificate file to your WIPAC home dir:

```
$ scp Downloads/usercred.p12
pub.icecube.wisc.edu:.globus/usercred.p12
```

File access: GridFTP & digital certificates

```
sub-1 ~ $ openssl pkcs12 -in .globus/usercred.p12 -nokeys | grep subject
Enter Import Password:
MAC verified OK
subject=/DC=org/DC=cilogon/C=US/O=Google/CN=Gonzalo Merino A38921
```

Copy-paste this certificate subject to your LDAP profile (the red string above)

https://internal.icecube.wisc.edu/directory/

File access: GridFTP & digital certificates

```
sub-1 ~ $ chmod 600 .globus/usercert.p12

sub-1 ~ $ voms-proxy-init

sub-1 ~ $ voms-proxy-info
subject   : /DC=org/DC=cilogon/C=US/O=Google/CN=Gonzalo Merino
A38921/CN=2882735090
issuer    : /DC=org/DC=cilogon/C=US/O=Google/CN=Gonzalo Merino A38921
identity  : /DC=org/DC=cilogon/C=US/O=Google/CN=Gonzalo Merino A38921
type      : RFC compliant proxy
strength  : 1024 bits
path      : /tmp/x509up_u38309
timeleft  : 11:59:54

sub-1 ~ $ export X509_USER_PROXY=/tmp/x509up_u$UID
```

Icetray-start wrapper script

http://software.icecube.wisc.edu/offline_trunk/metaproject/cvmfs.html

Self-contained IceTray Scripts

The CVMFS repository includes a wrapper script that you can use as the interpreter line in a Python script to automatically load an IceTray metaproject environment before execution. This makes the script easier to submit to a batch system while also documenting which metaproject it is intended to work with. To use it, put a line like the following at the top of your Python script:

```
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/py2-v1/icetray-start
#METAPROJECT XXXXX
```

The metaproject specification XXXXX can either be

- a build directory:

```
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/py2-v1/icetray-start
#METAPROJECT /data/user/you/metaprojects/icerec/build
```

- a tarball URL:

```
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/py2-v1/icetray-start
#METAPROJECT http://convey.icecube.wisc.edu/data/user/you/tarballs/icerec-trunk
```

- one of the *Pre-compiled metaprojects* distributed through the CVMFS repository

Pre-compiled metaprojects

The CVMFS repository includes pre-built copies of commonly used versions of the offline-software, icerec, and simulation metaprojects. To use a pre-built metaproject, put a line like the following at the top of your Python script:

```
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/py2-v1/icetray-start
#METAPROJECT: metaproject/VXX-YY-ZZ
```

Icetray Script (link to file)

```
sub-1 ~ $ cat icetray-start_script.py
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/py2-v2/icetray-start
#METAPROJECT icerec/V05-00-05

from argparse import ArgumentParser
from icecube import icetray, dataio
from I3Tray import *

parser = ArgumentParser()
parser.add_argument("-i", "--input", dest="INFILE")
parser.add_argument("-o", "--output", dest="OUTFILE")
args = parser.parse_args()

grid = 'gsiftp://gridftp-users.icecube.wisc.edu'
infile = grid + args.INFILE
outfile = grid + args.OUTFILE

tray = I3Tray()
tray.Add(dataio.I3Reader, Filename=infile)
tray.Add("Dump")
tray.Add("I3Writer", Filename=outfile)
tray.Execute(10)
```

Icetray Script

```
sub-1 ~ $ cat icetray-start_script.py
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/py2-v2/icetray-start
#METAPROJECT icerec/V05-00-05

from argparse import ArgumentParser
from icecube import icetray, dataio
from I3Tray import *

parser = ArgumentParser()
parser.add_argument("-i", "--input", dest="INFILE")
parser.add_argument("-o", "--output", dest="OUTFILE")
args = parser.parse_args()

grid = 'gsiftp://gridftp-users.icecube.wisc.edu'
infile = grid + args.INFILE
outfile = grid + args.OUTFILE

tray = I3Tray()
tray.Add(dataio.I3Reader, Filename=infile)
tray.Add("Dump")
tray.Add("I3Writer", Filename=outfile)
tray.Execute(10)
```

Then as usual …

```
sub-1 ~ $ chmod +x icetray-start_script.py
```

Submit many jobs with one submit file

HTCondor has built-in ways to submit multiple independent jobs with one submit file

Advantages

- Analyze multiple data files
- Test parameter combinations
- …

Without having to create submit files for each job, and submit them individually.

Icetray script submit file (link to file)

```
sub-1 ~ $ cat icetray.sub
executable = icetray-start_script.py
arguments = -i $(file) -o /data/user/gmerino/grid/$(Cluster)_$(Process)_$Fnx(file)

log = /scratch/gmerino/logs/$(Cluster).log
output = /scratch/gmerino/logs/$(Cluster)_$(Process).out
error = /scratch/gmerino/logs/$(Cluster)_$(Process).out

request_cpus = 1
request_memory = 1GB
request_disk = 2GB

Requirements = HAS_CVMFS_icecube_opensciencegrid_org

use_x509userproxy=true

queue file from job_files.txt
```

Icetray script submit file

```
sub-1 ~ $ cat icetray.sub
executable = icetray-start_script.py
arguments = -i $(file) -o /data/user/gmerino/grid/$(Cluster)_$(Process)_$Fnx(file)

log = /scratch/gmerino/logs/$(Cluster).log
output = /scratch/gmerino/logs/$(Cluster)_$(Process).out
error = /scratch/gmerino/logs/$(Cluster)_$(Process).out

request_cpus = 1
request_memory = 1GB
request_disk = 2GB

Requirements = HAS_CVMFS_icecube_opensciencegrid

use_x509userproxy=true

queue file from job_files.txt
```

Queue one job per entry in the job_files.txt file

Substitute $file variable when expanding macro

Icetray script submit file

```
sub-1 ~ $ cat icetray.sub
executable = icetray-start_script.py
arguments = -i $(file) -o /data/user/gmerino/grid/$(Cluster)_$(Process)_$Fnx(file)

log = /scratch/gmerino/logs/$(Cluster).log
output = /scratch/gmerino/logs/$(Cluster)_$
error = /scratch/gmerino/logs/$(Cluster)_$(

request_cpus = 1
request_memory = 1GB
request_disk = 2GB

Requirements = HAS_CVMFS_icecube_openscienc

use_x509userproxy=true

queue file from job_files.txt
```

$F() - useful macro for manipulating file names

$file = "./some/path/dir/filename.extension"

$Fp(file) -> ./some/path/dir/
$Fd(file) -> dir/
$Fn(file) -> filename
$Fx(file) -> .extension
$Fnx(file) -> filename.extension

Files list for the submit script (link to file)

```
sub-1 ~ $ cat job_files.txt
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000000.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000001.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000002.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000003.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000004.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000005.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000006.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000007.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000008.i3.bz2
/data/sim/IceCube/2012/filtered/level2/CORSIKA-in-ice/11926/00000-00999/Level2_IC86.2012_corsika.011926.000009.i3.bz2
```

Submit 10 file processing jobs

```
sub-1 ~ $ condor_submit icetray.sub
Submitting job(s)..........
10 job(s) submitted to cluster 13075424.

sub-1 ~ $ tail -f /scratch/gmerino/logs/13075424.log
```

```
cobalt07 ~/bootcamp $ ls  /data/user/gmerino/grid/13075424*
/data/user/gmerino/grid/13075424_0_Level2_IC86.2012_corsika.011926.000000.i3.bz2
/data/user/gmerino/grid/13075424_6_Level2_IC86.2012_corsika.011926.000006.i3.bz2
/data/user/gmerino/grid/13075424_1_Level2_IC86.2012_corsika.011926.000001.i3.bz2
/data/user/gmerino/grid/13075424_7_Level2_IC86.2012_corsika.011926.000007.i3.bz2
/data/user/gmerino/grid/13075424_2_Level2_IC86.2012_corsika.011926.000002.i3.bz2
/data/user/gmerino/grid/13075424_8_Level2_IC86.2012_corsika.011926.000008.i3.bz2
/data/user/gmerino/grid/13075424_4_Level2_IC86.2012_corsika.011926.000004.i3.bz2
/data/user/gmerino/grid/13075424_9_Level2_IC86.2012_corsika.011926.000009.i3.bz2
/data/user/gmerino/grid/13075424_5_Level2_IC86.2012_corsika.011926.000005.i3.bz2
```

Where did my jobs run?

condor_history command retrieves Class Ads for completed jobs. Accepts the same “autoformat” flag as condor_q

```
sub-1 ~ $ condor_history  13075424 -af LastRemoteHost

slot1@glidein_524087_132839132@cabinet-5-5-34.t2.ucsd.edu
slot1@glidein_2405516_480980500@cabinet-11-11-9.t2.ucsd.edu
slot1@glidein_2636731_630245952@red-c0821.unl.edu
slot1@glidein_1561967_43155000@cabinet-4-4-34.t2.ucsd.edu
slot1@glidein_752333_642282192@cabinet-8-8-6.t2.ucsd.edu
slot1@glidein_1563062_25053400@cabinet-4-4-34.t2.ucsd.edu
slot1@glidein_203500_32947863@cabinet-5-5-14.t2.ucsd.edu
slot1@glidein_2726240_44844460@cabinet-4-4-26.t2.ucsd.edu
slot1@glidein_466408_591189250@cabinet-4-4-16.t2.ucsd.edu
```

Running the same jobs in NPX

```
sub-1 ~ $ cat icetray.sub
executable = icetray-start_script.py
arguments = -i $(file) -o /data/user/gmerino/grid/$(Cluster)_$(Process)_$Fnx(file)

log = /scratch/gmerino/logs/$(Cluster).log
output = /scratch/gmerino/logs/$(Cluster)_$(Process).out
error = /scratch/gmerino/logs/$(Cluster)_$(Process).out

request_cpus = 1
request_memory = 1GB
request_disk = 2GB

Requirements = HAS_CVMFS_icecube_opensciencegrid_org

use_x509userproxy=true
should_transfer_files = YES
when_to_transfer_output = ON_EXIT

queue file from job_files.txt
```

Links

More tutorials can be found at:

- An Introduction to Using HTCondor
- Introduction to Workflows with DAGMan
- HTCondor advanced job submission

IceCube-specific information:

- Condor wiki

Exercise

To practice submitting to condor and doing real work, let's find all events that pass the min bias filter from the first 100 Level2 files in this directory:

```
/data/sim/IceCube/2011/filtered/level2/CORSIKA-in-ice/10668/01000-01999/
```

Some hints:

- A magic shebang is:

```
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/standard/icetray-start
#METAPROJECT: offline-software/trunk
```

- Be sure to check that the prescale passed too:

```
frame['FilterMask'][filter].condition_passed and
frame['FilterMask'][filter].prescale_passed
```

Exercise answer

Script to process each file:

```
#!/bin/sh /cvmfs/icecube.opensciencegrid.org/standard/icetray-start
#METAPROJECT: offline-software/trunk
import sys,os

input = sys.argv[1]
output = os.path.join(sys.argv[2],os.path.basename(input))

from icecube import dataclasses,dataio

outfile = dataio.I3File(output,'w')
try:
    for frame in dataio.I3File(input):
        if 'FilterMask' in frame and frame['FilterMask']['FilterMinBias_11']:
            outfile.push(frame)
finally:
    outfile.close()
```

Contact

Email:

- Problems/questions: help@icecube.wisc.edu
- discussion/information (mailing list): icecube-computing@icecube.wisc.edu

Slack channels:

- #icecube-it
- #software
- ...