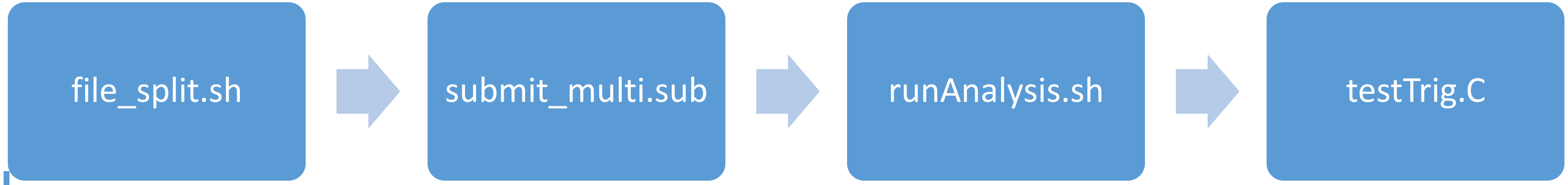
Condor job submit

Kung-Hsiang Chen

2022/6/27

https://github.com/AllenChen1997/condor_submit

Structure (if you follow the steps in README)



file_split.sh

submit_multi.sub

runAnalysis.sh

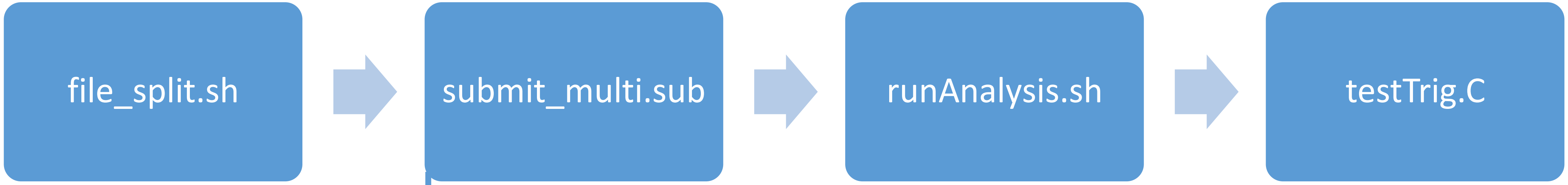
testTrig.C

Input:
JetHT-Run2017B-31Mar2018-v1_forTest.txt

Output:
signalList
JetHT-Run2017B-31Mar2018-v1_forTest_for_submit.txt
JetHT-Run2017B-31Mar2018-v1_forTest.log
submit_multi_temp.sub

Separate input file list into proper list size
Will call condor to submit jobs with submit_multi_temp.sub
The submit info. will save in JetHT-Run2017B-31Mar2018-v1_forTest.log

Structure (if you follow the steps in README)



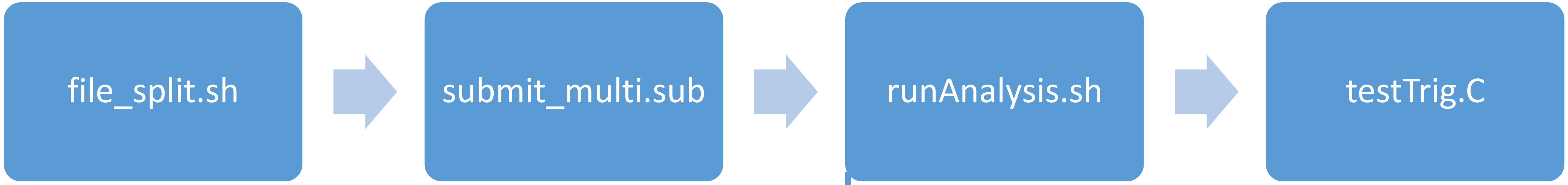
file_split.sh → submit_multi.sub → runAnalysis.sh → testTrig.C

Input:
JetHT-Run2017B-31Mar2018-v1_forTest_for_submit.txt

Output:
JetHT-Run2017B-31Mar2018-v1_forTest.log

Mainly control the configs about the condor job
This .sub file is the template file
When submit I prefer to copy and submit(submit_multi_temp.sub

Structure (if you follow the steps in README)



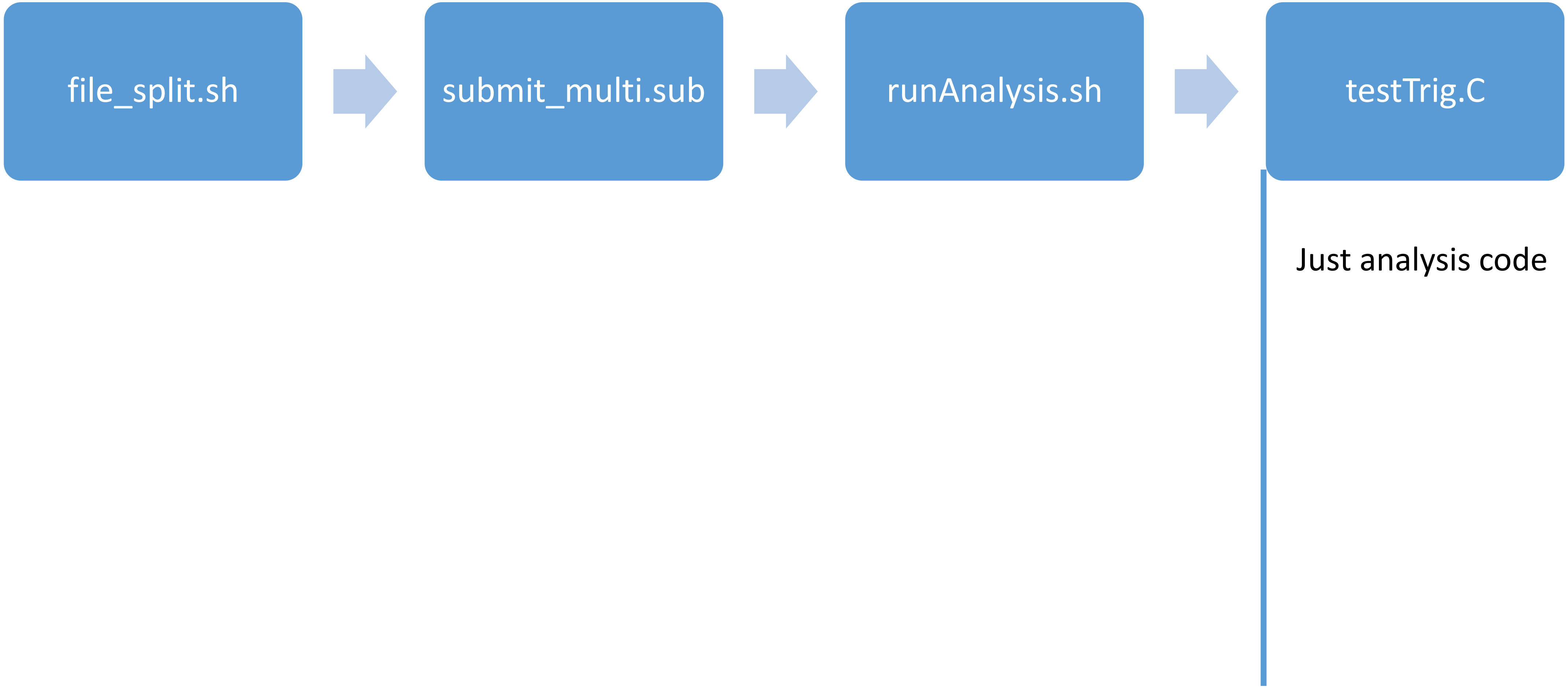
file_split.sh → submit_multi.sub → runAnalysis.sh → testTrig.C

Input:
- One line in the JetHT-Run2017B-31Mar2018-v1_forTest_for_submit.txt
- (Proxy file)
- outputname

Call the analysis code

Send the outputfile after finish the analysis code to the place you setup

Structure (if you follow the steps in README)



file_split.sh

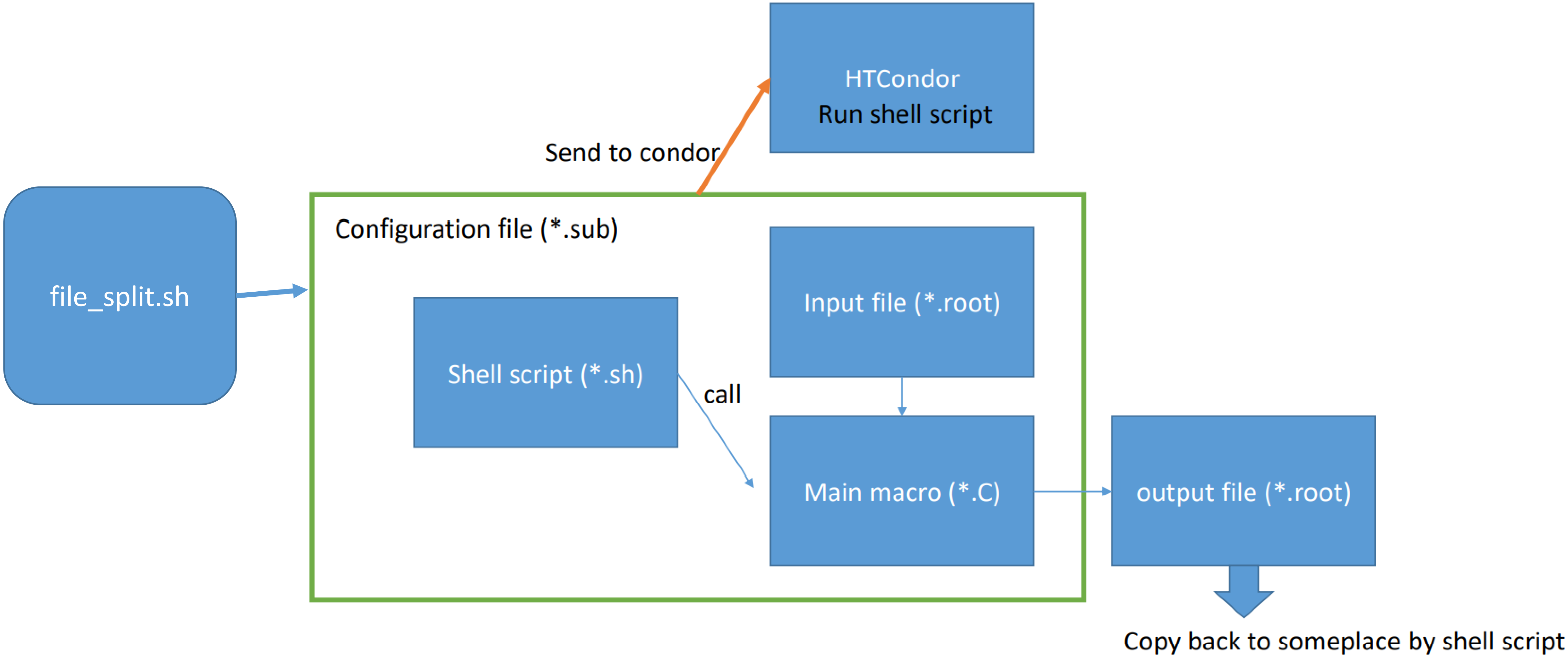
submit_multi.sub

runAnalysis.sh

testTrig.C

Just analysis code

Relation between codes



Prepare the config file

submit_multi.sub

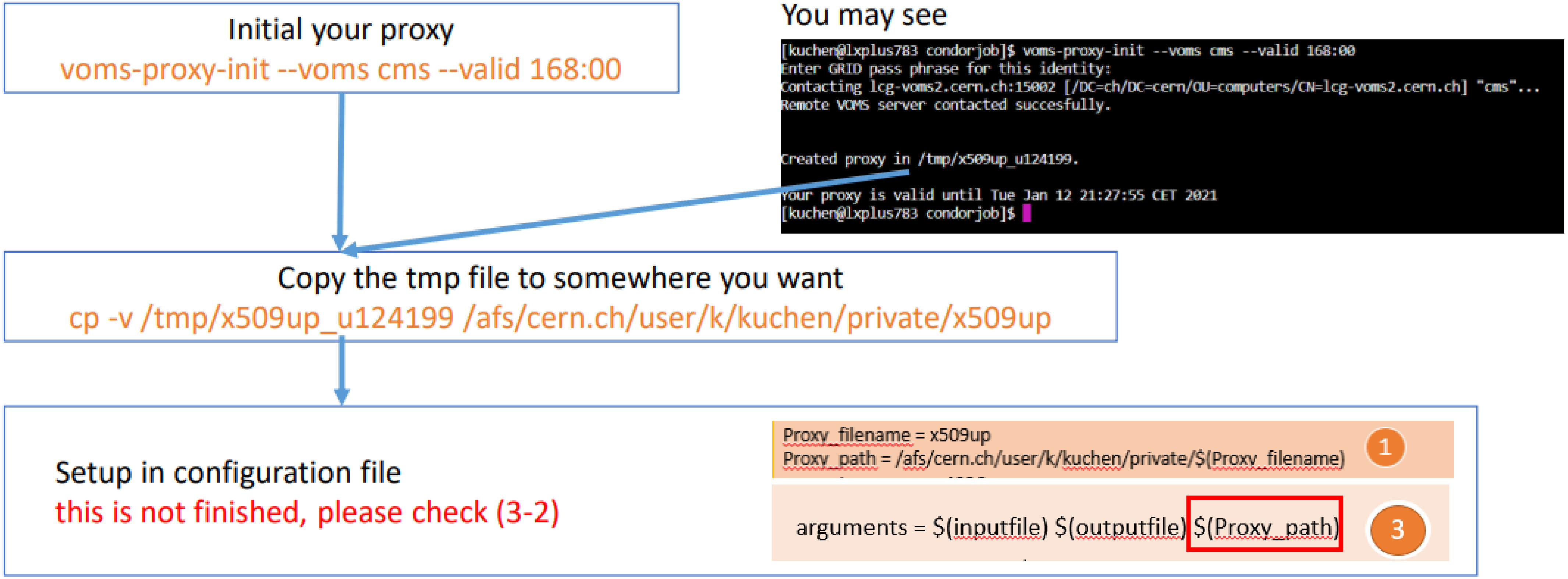
```
universe = vanilla

Proxy_filename = x509up        1

Proxy_path = /afs/cern.ch/user/k/kuchen/private/$(Proxy_filename)

#request_memory = 8192         2

#request_cpus = 4

+JobFlavour = "tomorrow"


listFile = tmplist.txt

outputname = testlist

outputfile = $(outputname)_$(Cluster)_$(Process).root


executable = runAnalysis.sh    3

should_transfer_files = YES

output = output/condor.$(Cluster).$(Process).out
```

```
error = error/condor.$(Cluster).$(Process).err

log = log/condor.$(Cluster).$(Process).log

transfer_input_files = runAnalysis.sh, testTrig.C, $(inputfile),
dummy.txt                                                          4

transfer_output_files = dummy.txt

.

on_exit_remove = (ExitBySignal == False) && (ExitCode == 0)        5

on_exit_hold = ( (ExitBySignal == True) || (ExitCode != 0) )

on_exit_hold_reason = strcat("Job held by ON_EXIT_HOLD due to
",ifThenElse((ExitBySignal == True), "exit by signal",strcat("exit code
",ExitCode)), ".")

periodic_release =  (NumJobStarts < 5) && ((CurrentTime -
EnteredCurrentStatus) > (60*60))


arguments = $(inputfile) $(outputfile) $(Proxy_path)    3

queue inputfile from $(listFile)
```

configuration file (1) setup proxy

- If the job needs proxy to access files in other place…

e.g.
root://cmsxrootd.hep.wisc.edu///store/user/khurana/ExoPieElement/setup_2017_2016_v06/TTToSemiLeptonic_TuneCP5_PSweights_13TeV-powheg-pythia8/crab_TTToSemiLeptonic_TuneCP5_PSweights_13TeV-powheg-pythia8/200523_000250/0000/ExoPieElementTuples_1-1.root



Initial your proxy

```
voms-proxy-init --voms cms --valid 168:00
```

You may see

```
[kuchen@lxplus783 condorjob]$ voms-proxy-init --voms cms --valid 168:00
Enter GRID pass phrase for this identity:
Contacting lcg-voms2.cern.ch:15002 [/DC=ch/DC=cern/OU=computers/CN=lcg-voms2.cern.ch] "cms"...
Remote VOMS server contacted succesfully.


Created proxy in /tmp/x509up_u124199.

Your proxy is valid until Tue Jan 12 21:27:55 CET 2021
[kuchen@lxplus783 condorjob]$
```

Copy the tmp file to somewhere you want

```
cp -v /tmp/x509up_u124199 /afs/cern.ch/user/k/kuchen/private/x509up
```

Setup in configuration file

this is not finished, please check (3-2)

```
Proxy_filename = x509up
Proxy_path = /afs/cern.ch/user/k/kuchen/private/$(Proxy_filename)
```

1

```
arguments = $(inputfile) $(outputfile) $(Proxy_path)
```

3

configuration file (2) job type

- In order to help scheduling or setup the requiring for the job, here are some options:
- request_memory = 4096
- request_cpus = 4
- +JobFlavour = "tomorrow"
  - This is the maximum job time. If the job is out of time, it will be terminated
  - Here is the table for Flavour

espresso	20min
microcentury	1h
longlunch	2h
workday	8h
tomorrow	1d
testmatch	3d
nextweek	1w

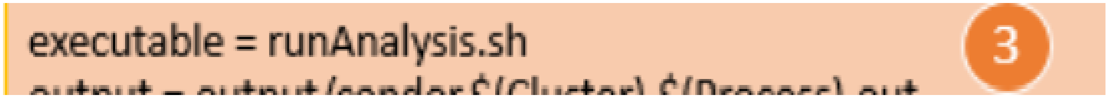
  - +MaxRuntime = Number of seconds also do the same thing

configuration file (3-1) executable code

- Write your own shell script to do the things
  - For example the bash script:

```
#!/bin/bash
root –b –q yourcode.C++\(\”var1\”,\”var2\”\)
python yourcode.py
xrdcp <outputfile> <the_dir_you_want>
```

- Add these in configuration file



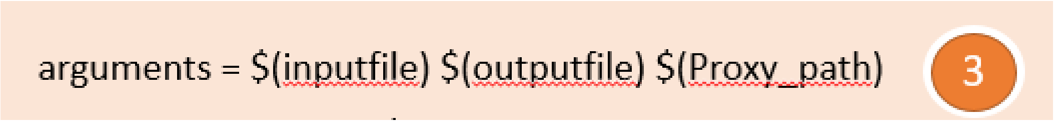
```
executable = runAnalysis.sh
output = output/condor.$(Cluster).$(Process).out
```

- Shell scripts can use some variable inputs when run as $n

  They are setup in arguments

  In this example $1 is tmplist.txt, $2 is tmplist.root, $3 is $(Proxy_path)

  For the usage example, you can check the next page

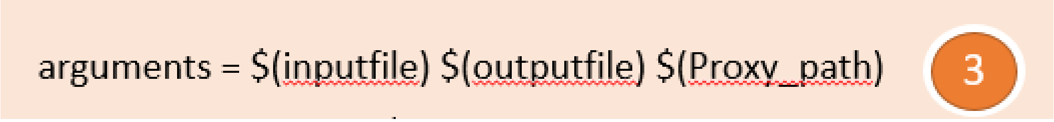


```
arguments = $(inputfile) $(outputfile) $(Proxy_path)
```

configuration file (3-2) executable code

- If you want to setup the proxy in condor

```
export X509_USER_PROXY=$3
voms-proxy-info -all
voms-proxy-info -all -file $3
```



```
arguments = $(inputfile) $(outputfile) $(Proxy_path)
```

In this example:

$1 is $(inputfile)
$2 is $(outputfile)
$3 is (Proxy_path)

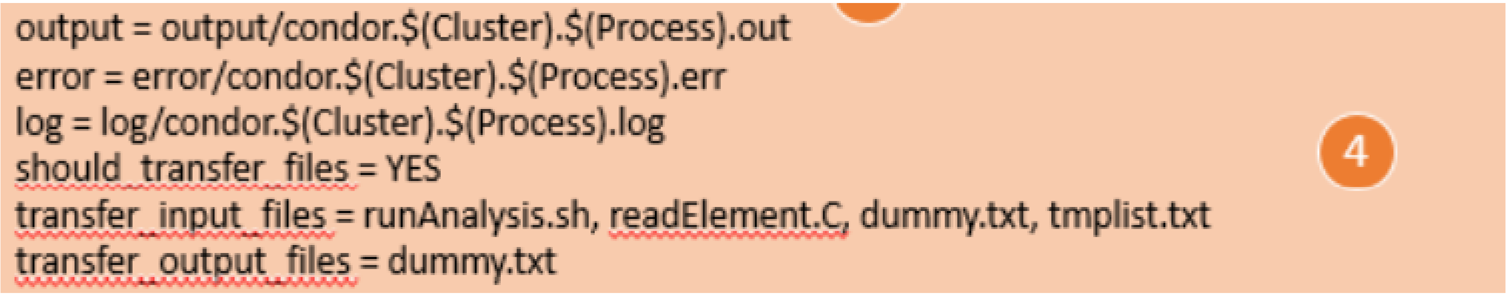
Combine previous page, you can run your root macro like this:

```
root –b –q yourcode.C++\(\”$1\”,\”$2\”\)
```

see runAnalysis.sh

configuration file (4) input / output files

- There are some log files can help you to know the status about the job
- output will collect the contents when running your shell scripts
- error will collect the errors when running your shell scripts
- log will collect the job status along the time
- transfer_input/output_files declare the files to transfer to/from condor (make sure the file exist)

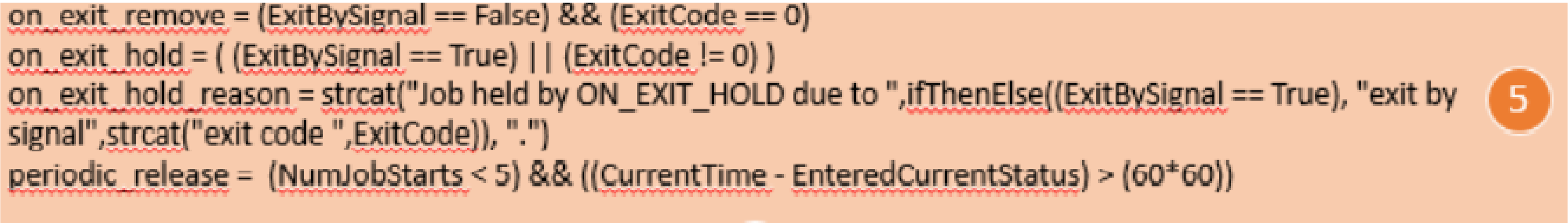


```
output = output/condor.$(Cluster).$(Process).out
error = error/condor.$(Cluster).$(Process).err
log = log/condor.$(Cluster).$(Process).log
should_transfer_files = YES
transfer_input_files = runAnalysis.sh, readElement.C, dummy.txt, tmplist.txt
transfer_output_files = dummy.txt
```

The relative directory path to file is also ok, e.g. ../../somefile.txt

configuration file (5) job state control

- on_exit_remove: if the condition is true, it leave the job queue normally. If false, placed back into the Idle state
- on_exit_hold: if the condition is true, it place job into hold state. If false, nothing happened
- on_exit_hold_reason: show the description when on_exit_hold is true
- periodic_release: it the condition is true, the job will be released



```
on_exit_remove = (ExitBySignal == False) && (ExitCode == 0)
on_exit_hold = ( (ExitBySignal == True) || (ExitCode != 0) )
on_exit_hold_reason = strcat("Job held by ON_EXIT_HOLD due to ",ifThenElse((ExitBySignal == True), "exit by
signal",strcat("exit code ",ExitCode)), ".")
periodic_release =  (NumJobStarts < 5) && ((CurrentTime - EnteredCurrentStatus) > (60*60))
```

Quick resubmit

- In case sometime you may meet issue when the serve is not stable. S.t the input file can not open…
- I write a short shell script which can help to find out the condor jobs which fail

resubmit_list_check.sh

There are two key point to make it possible:

- there are “Error” words in output/*.out

- runAnalysis.sh: echo “1 is $1” is needed, make sure not comment it

Useful commands

- condor_submit yourconfig.sub submit the job
- condor_q check the job state
- condor_q –analyze <jobid>   show more details

e.g. condor_q –analyze 7082186.0

- condor_tail <jobid>   if the state in run, show the contents in output now
- condor_rm <jobid>   remove the job
- condor_rm <your user name> remove all job you submit

Other links

- Basic concept for Htcondor:

https://indico.cern.ch/event/611296/contributions/2604376/attachments/1471164/2276521/TannenbaumT_UserTutorial.pdf

- Some introduction about content of configuration file:

https://batchdocs.web.cern.ch/local/submit.html

condor_submit (wisc.edu)