

Frameworks Homework

IF Computing Summer School

Problem 1 Working with UPS

1. Login to a gpvm and setup the software for your experiment
2. Look at all of the products that are now set up by running the UPS command

```
ups active
```

You should see a long list of product names, with the version, flavor, qualifiers, and location (-z /cvmfs/...) following the name

3. Confirm that the executable for your experiment is listed (e.g. gm2 for g-2, mu2e for Mu2e, ...) by piping the name of the executable through 'grep'

```
ups active | grep gm2
```

4. Look at the help for your experiment's executable with the command

```
<executable_name> -h
```

You should see something like this

```
Usage: gm2 <-c <config-file>> <other-options> [<source-file>]+
```

followed by a listing of options, and ending with this

```
Art has completed and will exit with status 1.
```

Problem 1 cont.

5. Observe the dependencies of a particular product using the command

```
ups depend <product_name> <version> -f <flavor> -q <qualifiers>
```

Remember, `ups active` will show the version, flavor and qualifiers of a product so you can copy those from that output.

You should see a dependency graph that looks something like this:

```
-bash-4.2$ ups depend geant4 v4_10_3_p03agm2 -f Linux64bit+3.10-2.17 -q e15:prof:us
geant4 v4_10_3_p03agm2 -f Linux64bit+3.10-2.17 -z /cvmfs/gm2.opensciencegrid.org/prod/external -q e15:prof:us
|__clhep v2_3_4_5c -f Linux64bit+3.10-2.17 -z /cvmfs/gm2.opensciencegrid.org/prod/external -q e15:prof
|  |__gcc v6_4_0 -f Linux64bit+3.10-2.17 -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__xerxes_c v3_1_4b -f Linux64bit+3.10-2.17 -z /cvmfs/gm2.opensciencegrid.org/prod/external -q e15:prof
|__vecgeom v0_4a -f Linux64bit+3.10-2.17 -z /cvmfs/gm2.opensciencegrid.org/prod/external -q e15:prof
|__g4abla v3_0 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4emlow v6_50 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4neutron v4_5 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4neutronxs v1_4 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4nucleonxs v1_1 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4nuclide v2_1 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4photon v4_3_2 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4pii v1_3 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4radiative v5_1_1 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4surface v1_0 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
|__g4tendl v1_3 -f NULL -z /cvmfs/gm2.opensciencegrid.org/prod/external
-bash-4.2$
```

This is the dependency graph for `geant4 v4_10_3_p02agm2` for an SL7 build (Linux64bit+...) with qualifiers `e15:prof:us`. Notice that this 'version' of `geant4` depends on `clhep v2_3_4_5c ...`, `xerxes_c v3_1_4b ...`. In turn, `clhep` depends on `gcc v6_4_0`. These dependency graphs are pretty complicated for the experimental executables themselves.

Problem 1 cont.

6. Look at all of the instances of *art* that are available to you with the command

```
ups list -aK+ art
```

You should see a list of several (perhaps many) options all with different versions, flavors, and qualifiers

7. Find the location on cvmfs of the instance of art that is setup and do an `'ls'` on that directory to list its contents, eg for g-2

```
ls /cvmfs/gm2.opensciencegrid.org/prod/external/art
```

You will see several (perhaps many) directories with names like `v2_10_01` and `v2_10_01.version`.

8. Check out the contents of those directories.

➔ The non-.version directories have the source code, the includes (header files), a `ups` directory with the `.table` file in it, and at least one directory with a name like `slf7.x86_64.e15.prof`, which contains the binary files in `/lib` and `/bin`.

➔ The .version directories have at least one file in them with a name like `Linux64bit+2.6-2.12_e15_prof`. You can see the contents using `'cat'` or `'less'` on these files. These are needed by UPS.

Problem 1 cont.

9. Every product installed in your work area will have environment variables defined for the locations of the main directory (eg \$ART_DIR), the lib directory (eg \$ART_LIB), the include directory (eg \$ART_INC), as well as other useful definitions. Use the 'echo' command to see these definitions:

`echo $ART_` (follow with tab to see the full listing of env variables starting with ART_)

One example is \$ART_LIB, `echo $ART_LIB` Output for me is:

```
/cvmfs/gm2.opensciencegrid.org/prod/external/art/v2_10_03/slf7.x86_64.e15.prof/lib
```

Look at \$ART_VERSION with `echo $ART_VERSION` No surprise there! I get

```
v2_10_03
```

Don't forget that these environment variables are set once a product is setup in your work area. They can be useful!

Problem 2 Transferring Data

1. Login to a gpvm and setup the software for your experiment
2. Find out what type of storage volumes are available on a node using

```
df -h
```

You should see a long list of storage volumes, with information about the total size, available size, mount point, device location.

3. Setup the Intensity Frontier Data Handling Client (ifdhc) using UPS:

```
setup ifdhc v2_5_4
```

4. Copy a file (to your /dev/null directory - look up dev/null on the internet) using the client:

```
ifdh cp root://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/dune/tape_backed/ dunepro/  
physics/full-reconstructed/2019/mc/out1/PDSPProd2/22/60/37/10/  
PDSPProd2_protoDUNE_sp_reco_35ms_sce_off_23473772_0_452d9f89- a2a1-4680-  
ab72-853a3261da5d.root /dev/null
```

You will not see the file in the /dev/null directory. If you looked up dev/null you will know why.

Problem 2 cont.

5. Create a directory in your dCache scratch area, /pnfs/<your_experiment_name>/scratch/users/{USER}/ called “IFC_Summer_School”
6. Copy your ~/.bashrc to that directory using `ifdhc cp`
7. Copy the .bashrc file from your IFC_Summer_School dCache directory to /dev/null using the client
8. Remove the directory /pnfs/<your_experiment_name>/scratch/users/{USER}/IFC_Summer_School using the client:

```
ifdh rmdir /pnfs/<your_experiment_name>/scratch/users/{USER}/  
IFC_Summer_School
```

Note: to delete a directory using the ifdh client, the directory must be empty.

See https://cdcvs.fnal.gov/redmine/projects/ifdhc/wiki/lfdh_commands for details on the ifdhc commands.

Problem 2 cont.

9. ROOT provides a client-server system which allows the reading of ROOT files over a network. Since it is root-aware, it can read branches and baskets in an optimal way. It only reads what it needs, so if you only read a small part of the file or event, this may save time in transferring data. Issue the following commands. Try to understand how the first command enables completing the parameters for the second command.

```
pnfs2xrootd /pnfs/<your_experiment>/scratch/users/${USER}/
```

You should get an output like this:

```
root://fndca1.fnal.gov:1094/pnfs/fnal.gov/usr/<your_experiment>/scratch/users/  
<your_username>
```

(Note, if your experiment does not support the `pnfs2xrootd` command then you should know that in order to use `xrootd` the file path must be in canonical dCache form, i.e. `/pnfs/mu2e/...` **takes the form** `xroot://fndca1.fnal.gov/pnfs/fnal.gov/usr/mu2e/...`

```
xrdafs root://fndca1.fnal.gov:1094/ ls /pnfs/fnal.gov/usr/<your_experiment>/  
scratch/users/${USER}
```

How many directories/files do you see in there? Checkout all of the `xrdafs` capabilities by simply typing the command `'xrdafs'` or `'xrdafs --help'` to see the help.

Problem 3 Using SAM

1. Login your experiment general purpose virtual machine
 - ssh -XYK twalton@gm2gpvm02.fnal.gov
2. Setup your experiment software and common area
 1. source /cvmfs/gm2.opensciencegrid.org/prod/g-2/setup
 2. source /grid/fermiapp/products/common/etc/setups.sh
3. Setup SAM
 1. setup sam_web_client
 2. export SAM_EXPERIMENT=gm2

Problem 3 cont.

- Check you have a valid kerberos ticket on the gpvm machine

```
[<gm2gpvm04.fnal.gov> klist
Ticket cache: FILE:/tmp/krb5cc_43472_VMqK41iG3X
Default principal: twalton@FNAL.GOV

Valid starting    Expires          Service principal
06/17/2021 20:40:57 06/18/2021 22:40:21  krbtgt/FNAL.GOV@FNAL.GOV
06/17/2021 20:40:57 06/18/2021 22:40:21  nfs/homesrv01.fnal.gov@FNAL.GOV
06/17/2021 20:40:57 06/18/2021 22:40:21  nfs/filesrv01.fnal.gov@FNAL.GOV
```

- Invalid tickets

```
[<gm2gpvm04.fnal.gov> kinit twalton
[Password for twalton@FNAL.GOV:
<gm2gpvm04.fnal.gov> █
```

- To obtain authentication, a kx509 certificate is generated to communicate with the samweb server.

```
[<gm2gpvm04.fnal.gov> kx509
Checking if /tmp/x509up_u43472 can be reused ... no
Authorizing ..... authorized
Fetching certificate ..... fetched
Storing certificate in /tmp/x509up_u43472
Your certificate is valid until: Thu Jun 24 21:46:26 2021
```

Problem 3 cont.

SAM Authentication

- To test your authentication

```
[<gm2gpvm04.fnal.gov> samweb -s server-info  
SAMWeb API for gm2  
SAMWeb version: 3.1.0-pre1-15-g4188d8d  
Connected to: https://samweb.fnal.gov:8483/sam/gm2/api
```

```
CherryPy version: 18.6.0  
Backend server hostname: samwebgpvm06.fnal.gov  
SQLAlchemy version: 1.3.19  
Database: postgresql+psycopg2://samdbs:***@sampgsdb03.fnal.gov:5437/sam_gm2_prd  
HTTP User-Agent: SAMWebClient/v3_0 (samweb.py) python/2.7.5
```

User information:

```
Untrusted identity: twalton@gm2gpvm04.fnal.gov  
Authenticated username: twalton  
Roles: dataset, project, metadata
```

The received message for an authorized access

Problem 3 cont.

samweb : Help Menu

```
[<gm2gpvm04.fnal.gov> samweb --help
Usage: samweb.py [base options] <command> [command options] ...

Options:
  --version          show program's version number and exit
  -h, --help        show this help message and exit
  --help-commands   list available commands

Base options:
  -e EXPERIMENT, --experiment=EXPERIMENT
                        use this experiment server. If not set, defaults to
                        $SAM_EXPERIMENT if set, else $EXPERIMENT.
  --dev              use development server
  -s, --secure       always use secure (SSL) mode
  --cert=CERT        x509 certificate for authentication. If not specified,
                        use $X509_USER_PROXY, $X509_USER_CERT/$X509_USER_KEY
                        or standard grid proxy location
  --key=KEY          x509 key for authentication (defaults to same as
                        certificate)
  --max-timeout=MAX_TIMEOUT
                        set the max timeout in seconds (After this period even
                        retrieable errors will fail. This setting will not
                        interrupt an active connection; use socket-timeout for
                        that.)
  --socket-timeout=SOCKET_TIMEOUT
                        set the socket timeout in seconds (max time for data
                        to be sent or received)
  -r ROLE, --role=ROLE
                        specific role to use for authorization
  -z TIMEZONE, --timezone=TIMEZONE
                        set time zone for server responses
  -v, --verbose      Verbose mode
```

Problem 3 cont.

Investigate the Data Tier Categories for your experiment

```
[<gm2gpvm04.fnal.gov> samweb list-values --help_categories -e dune
Available database categories:
data_tiers
    Data tier of a file (raw, simulated, reconstructed, etc)
checksum_types
    Checksum type
run_types
    Type of a run
groups
    Work groups
data_streams
    Data stream names
file_formats
    Format of the file
file_types
    Type (ie general purpose) of a file
```

```
<gm2gpvm04.fnal.gov> samweb list-values data_tiers -e dune
simulated
raw
hit-reconstructed
full-reconstructed
generated
detector-simulated
reconstructed-2d
reconstructed-3d
sliced
dclinput
dcloutput
root-tuple
root-hist
dqm
decoded-raw
sam-user
pandora_info
reco-recalibrated
<gm2gpvm04.fnal.gov> samweb list-values data_tiers -e gm2
raw
calibration
fcl
log
reco
sim
analysis
nmr-test
mdc0
truth
processed
nearline
unpack
hists
sam-user
full
infilldqc
gain_corr
hadd
gaincorr
oof_corr
dqc_hadd
gain_hadd
qdb
subrun_dqc
qdb_dqc
```

Problem 3 cont.

Use the SAM Webserver to find your experiment's metadata parameters

<https://samweb.fnal.gov:8483/sam/nova/api/files/list/dimensions>

<https://samweb.fnal.gov:8483/sam/gm2/api/files/list/dimensions>

<https://samweb.fnal.gov:8483/sam/uboone/api/files/list/dimensions>

<https://samweb.fnal.gov:8483/sam/minerva/api/files/list/dimensions>

<https://samweb.fnal.gov:8483/sam/mu2e/api/files/list/dimensions>

<https://samweb.fnal.gov:8483/sam/<experiment>/api/files/list/dimensions>

Problem 3 cont.

Use metadata to create a SAM dataset

- Before creating a dataset, use “samweblist-files” to provide a summary of files in a potential dataset. **samweb list-files -h** will give you the help menu

```
<dunegpvm05.fnal.gov> samweb list-files -e gm2 "user gm2pro and file_format artroot and offline.requestid 5205P and run_number >= 31368 and run_number <= 32035" --summary
File count:      100850
Total size:      104491740150658
Event count:     21331213
<dunegpvm05.fnal.gov> samweb list-files -e gm2 "user gm2pro and file_format artroot and offline.requestid 5205P and run_number >= 31368 and run_number <= 32035" --dump-query
Parse tree for:

user gm2pro and file_format artroot and offline.requestid 5205P and run_number >= 31368 and run_number <= 32035

AndNode
  Dimension(user = gm2pro)
  Dimension(file_format = artroot)
  Dimension(offline.requestid = 5205P)
  Dimension(run_number >= 31368)
  Dimension(run_number <= 32035)

Normalized query:

user gm2pro and file_format artroot and offline.requestid 5205P and run_number >= 31368 and run_number <= 32035

PostgreSQL query:

SELECT DISTINCT ON (data_files.file_id) data_files.file_id, data_files.file_name
FROM data_files JOIN persons AS create_user ON data_files.create_user_id = create_user.person_id JOIN file_formats ON file_formats.file_format_id = data_files.file_format_id JOIN data_files_param_values AS dfpv_offline_requestid ON data_files.file_id = dfpv_offline_requestid.file_id JOIN data_files_runs ON data_files.file_id = data_files_runs.file_id JOIN runs ON runs.run_id = data_files_runs.run_id
WHERE create_user.username = 'gm2pro' AND file_formats.file_format = 'artroot' AND dfpv_offline_requestid.param_value_id = 16410519 AND runs.run_number >= 31368 AND runs.run_number <= 32035 AND data_files.retired_date IS NULL AND data_files.file_content_status_id = 1 AND (EXISTS (SELECT 1
FROM data_file_locations
WHERE data_file_locations.file_id = data_files.file_id))
```

Problem 3 cont.

Use metadata to create a SAM dataset

- Use the samweb command **samweb create-definition** to create a dataset.

```
– samweb create-definition gm2pro_daq_preproduction_Run3E_5211P "user gm2pro and file_format artroot and offline.requestid 5211P and run_number >= 32126 and run_number <= 32319"
```

- Count how many files are in the dataset.

```
[gm2pro@gm2gpvm01:~]> samweb count-definition-files gm2pro_daq_preproduction_Run3E_5211P  
32673
```

- Get the dataset description.

```
[gm2pro@gm2gpvm01:~]>  
[gm2pro@gm2gpvm01:~]> samweb describe-definition gm2pro_daq_preproduction_Run3E_5211P  
Definition Name: gm2pro_daq_preproduction_Run3E_5211P  
Definition Id: 362145  
Creation Date: 2021-06-15T17:14:48+00:00  
Username: gm2pro  
Group: gm2  
Dimensions: user gm2pro and file_format artroot and offline.requestid 5211P and run_number >= 32126 and run_number <= 32319  
[gm2pro@gm2gpvm01:~]>
```

Problem 3 cont.

Checkout the online Definition Editor for your experiment at

https://samweb.fnal.gov:8483/sam/<your_experiment_name>/definition_editor/

The screenshot shows a web browser window with the URL `https://samweb.fnal.gov:8483/sam/dune/definition_editor/`. The page title is "Dataset Definition Editor".

The interface is divided into several sections:

- Selection Criteria:** This section contains several input fields and buttons:
 - Data Set:** Includes a button "Add Reference to Dataset" and a dropdown menu "Show Datasets for Group" with "dune" selected.
 - Data Tier:** Includes a button "Add Data Tier" and a dropdown menu with "simulated" selected.
 - Version:** Includes a button "Add Version" and an empty input field.
 - Time Range:** Includes a button "Add Time Range", a "Start Time" dropdown, and an "to" dropdown. Below it, a note says "(format: 2011-05-09 or 2011-05-09T23:46:04)".
 - Run Number:** Includes a button "Add Run", a dropdown menu, and an empty input field.
 - Other:** Includes a button "Add Quantity", a dropdown menu with "Events" selected, and another dropdown menu.
- Operators:** A vertical sidebar on the right contains three buttons: "AND", "OR", and "S".
- Data Set Definition (Dimensions query):** A section with a text area for a query string. Below the text area are two buttons: "Clear Query" and "Get File List".
- Save Definition:** A section with three input fields: "Dataset Name", "User", and "Group" (with "dune" entered). Below these is a "Save" button.

At the bottom of the "Save Definition" section, there is a note: "Datasets can have an arbitrary name but should not include spaces or special characters (underscores and dashes are permitted)".

Problem 3 cont.

Review Tammy's talk pages 26-29 to remind yourself about presaging data from tape to disk before using it in a grid job.