TextFileGen Client/Server

Larsoft Coordination Meeting
May 31, 2022

H. Greenlee

Overview

- TextFileGen is an art producer module that reads events from a hepevt format text file and adds mc truth data products to the event.
- A major problem with running TextFileGen in batch jobs is that it has no built-in way of ensuring that different batch jobs receive different hepevt events.
  - Various ad-hoc solutions have been invented, which require support from workflow scripts to split the original input hepevt file into pieces.
    - These solutions work OK, provided batch jobs have equal number of events.
    - Not convenient for overlay MC.
- An alternative solution is presented in this talk, which is using a hepevt server to supply a different unique event each time it is accessed.
  - Solves the overlay/unequal-event-number problem.
  - Removes the need for external support from the work flow.

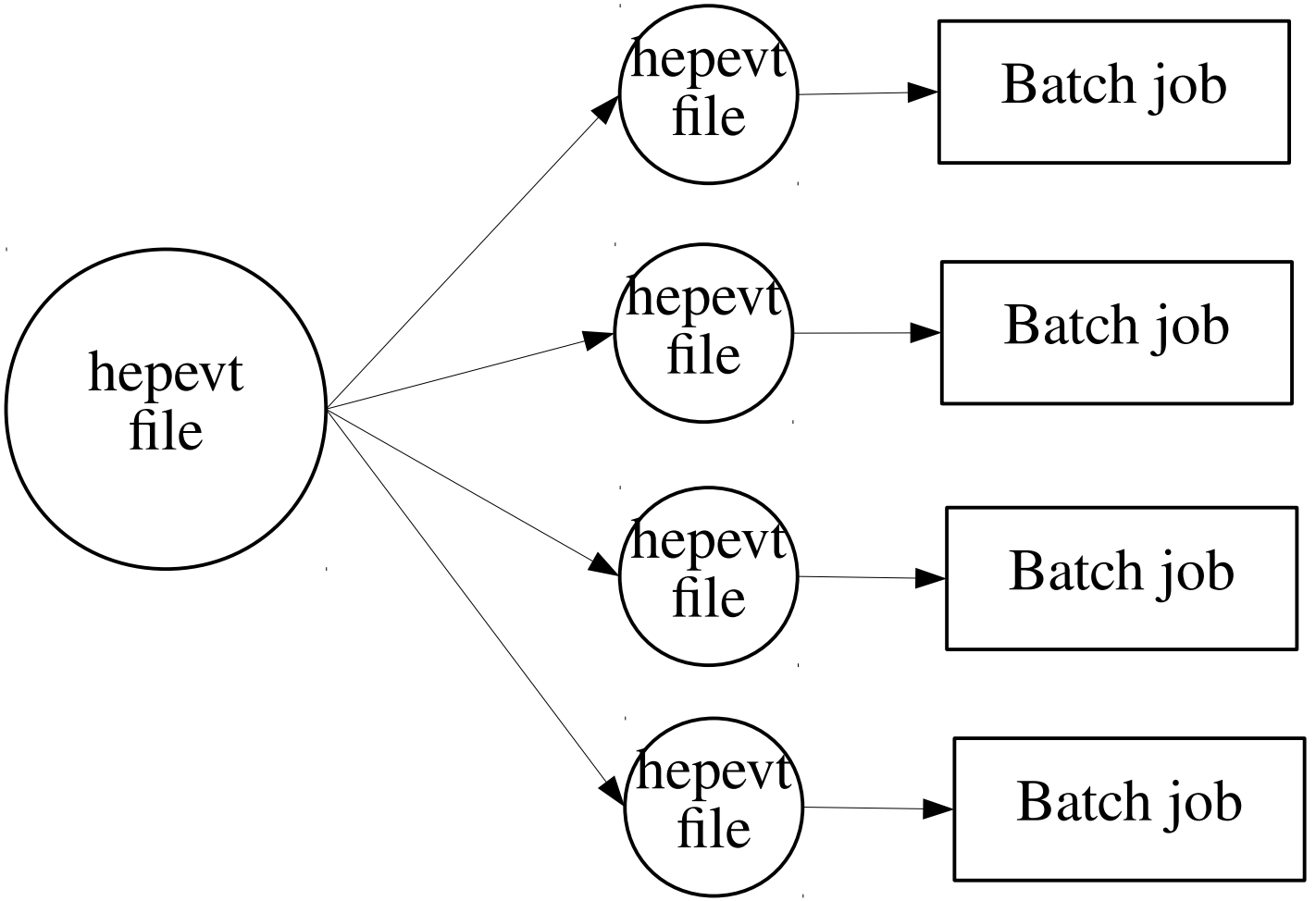
A Recent Update to TextFileGen

- TextFileGen was updated in v09_42_00 by Andrzej Szelc to add an event offset fcl parameter (larsim#85).
- This update gives a new tool to writers of workflow scripts supporting TextFileGen, but doesn't solve the fundamental issues with TextFileGen.

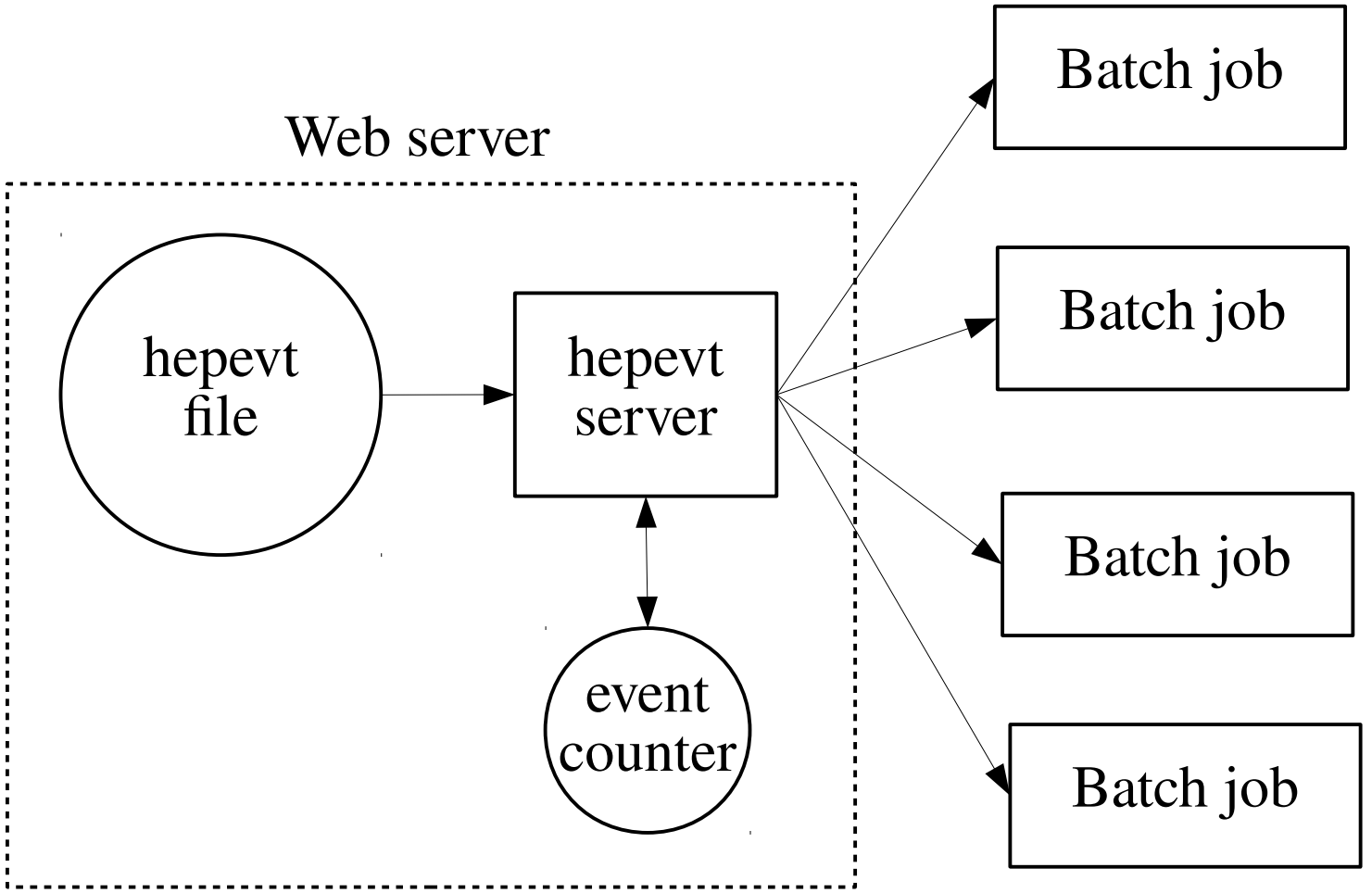
Outline

- TextFileGen server.
- TextFileGen client (TextFileGen module).
- Security & authentication.
- Scalability and performance.

Current Method:
TextFileGen Using File Splitting



New Method:
TextFileGen Client/Server



Hepevt Server

- Hepevt server is a cgi script installed on a web server (e.g. hosted on Fermilab central web hosting).
  - https://microboone-exp.fnal.gov/cgi-bin/hepevt.py
  - /web/sites/m/microboone-exp.fnal.gov/cgi-bin/hepevt.py
- Input hepevt file should be stored in data area of web site.
  - /web/sites/m/microboone-exp.fnal.gov/data/hepevt
- Basic features of hepevt.py script.
  - Each invocation returns one event as plain text.
  - Script reads and updates an event counter file, stored on server.
  - Understands hepevt and hepmc file formats.
  - Script can be invoked on command line, or via http(s).
  - Not experiment-specific.

Hepevt.py Example Invocations

```
$ hepevt.py -f HEPevents.txt
Content-type: text/plain
Status: 200 OK

379  2
1  11  0  0  0  0  0.0264716  -0.0274087  0.0934192  0.100893  0.000511  -41.5954  117.698  172.765  50
1  -11  0  0  0  0  0.261785  0.0595077  0.580473  0.639548  0.000511  -41.5954  117.698  172.765  50
$ curl 'https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?file=HEPevents.txt'
380  2
1  11  0  0  0  0  0.550249  0.0745048  1.15614  1.28257  0.000511  179.797  -150.629  -287.17  50
1  -11  0  0  0  0  0.541217  0.0378625  1.3602  1.46441  0.000511  179.797  -150.629  -287.17  50
```

Hepevt.py Example Invocations

First event, from shell.

```
$ hepevt.py -f HEPevents.txt
Content-type: text/plain
Status: 200 OK

379  2
1  11  0  0  0  0  0.0264716  -0.0274087  0.0934192  0.100893  0.000511  -41.5954  117.698  172.765  50
1  -11  0  0  0  0  0.261785  0.0595077  0.580473  0.639548  0.000511  -41.5954  117.698  172.765  50
$ curl 'https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?file=HEPevents.txt'
380  2
1  11  0  0  0  0  0.550249  0.0745048  1.15614  1.28257  0.000511  179.797  -150.629  -287.17  50
1  -11  0  0  0  0  0.541217  0.0378625  1.3602  1.46441  0.000511  179.797  -150.629  -287.17  50
```

Hepevt.py Example Invocations

```
$ hepevt.py -f HEPevents.txt
Content-type: text/plain
Status: 200 OK

379  2
1  11  0  0  0  0  0.0264716  -0.0274087  0.0934192  0.100893  0.000511  -41.5954  117.698  172.765  50
1  -11  0  0  0  0  0.261785  0.0595077  0.580473  0.639548  0.000511  -41.5954  117.698  172.765  50
$ curl 'https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?file=HEPevents.txt'
380  2
1  11  0  0  0  0  0.550249  0.0745048  1.15614  1.28257  0.000511  179.797  -150.629  -287.17  50
1  -11  0  0  0  0  0.541217  0.0378625  1.3602  1.46441  0.000511  179.797  -150.629  -287.17  50
```

http header

Hepevt.py Example Invocations

hepevt event data

```
$ hepevt.py -f HEPevents.txt
Content-type: text/plain
Status: 200 OK

379  2
1  11  0  0  0  0  0.0264716  -0.0274087  0.0934192  0.100893  0.000511  -41.5954  117.698  172.765  50
1  -11  0  0  0  0  0.261785  0.0595077  0.580473  0.639548  0.000511  -41.5954  117.698  172.765  50
$ curl 'https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?file=HEPevents.txt'
380  2
1  11  0  0  0  0  0.550249  0.0745048  1.15614  1.28257  0.000511  179.797  -150.629  -287.17  50
1  -11  0  0  0  0  0.541217  0.0378625  1.3602  1.46441  0.000511  179.797  -150.629  -287.17  50
```

Hepevt.py Example Invocations

```
$ hepevt.py -f HEPevents.txt
Content-type: text/plain
Status: 200 OK

379  2
1  11  0  0  0  0  0.0264716  -0.0274087  0.0934192  0.100893  0.000511  -41.5954  117.698  172.765  50
1  -11  0  0  0  0  0.261785  0.0595077  0.580473  0.639548  0.000511  -41.5954  117.698  172.765  50
$ curl 'https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?file=HEPevents.txt'
380  2
1  11  0  0  0  0  0.550249  0.0745048  1.15614  1.28257  0.000511  179.797  -150.629  -287.17  50
1  -11  0  0  0  0  0.541217  0.0378625  1.3602  1.46441  0.000511  179.797  -150.629  -287.17  50
```

second event, via https

Invoking hepevt.py

- Hepevt.py accepts arguments passed via command line, or via url.

```
$ hepevt.py -h
hepevt.py <arguments>

CLI arguments (for testing):

-h|--help                 - Print help message.
-f|--file <hepevt-file> - HepEvt file.
--format <format>         - File format, "hepevt" or "hepmc" (default "hepevt").
-s|--stream <stream>      - Event counter stream name (default='default').
-u|--user <user>          - User name (default none or SSO userid).
-e|--event <event>        - Specify event number.
-r|--reset                - Reset event counter for the specified stream.
--sleep <secs>            - Sleep time per event (default 0)
--min_event <event>       - Minimum event number.
--max_event <event>       - Maximum event number.

CGI arguments:

file      - Name of hepevt file (just file name, not path).
format    - File format, "hepevt" or "hepmc" (default "hepevt").
stream    - Event counter stream name (default='default').
user      - User name (default none or SSO userid).
event     - Specify event number.
reset     - Event counter reset flag (reset if "1").
sleep     - Sleep time per event.
min_event - Minimum event number.
max_event - Maximum event number.
```

Invoking hepevt.py (cont.)

- Invoking hepevt.py from a shell will only work if you have write access (content editor access) to the web server (mainly for debugging and development).
- Non-privileged users (including TextFileGen) can invoke hepevt.py by reading a specially formatted url.
  - CGI arguments are added at the end of the url:
    - `https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?name=value&name=value`
- Example, resetting the event counter.

```
$ curl 'https://microboone-exp.fnal.gov/cgi-bin/hepevt.py?file=HEPevents.txt&reset=1'
Reset event counter.
```

- Argument “file” is always required.

The Event Counter File

- The event counter file is always stored in the same directory as the hepevt data file.
- The event counter file stores two integers:
  - Event number of the last read event.
  - Byte offset of the last read event.
- The byte offset allows hepevt.py to rapidly seek to the approximate correct position in the file on subsequent invocations, so that hepevt.py does not have to read the entire file on each invocation.
- The event counter file persists indefinitely, until it is manually reset or deleted.
- Hepevt.py assumes event numbers are increasing, but event numbers don't have to be continuous (i.e. filtered hepevt files are OK).

The Event Counter File (cont.)

- Because there can be multiple hepevt.py processes trying to access the event counter file at the same time, hepevt.py uses posix file locking to obtain an exclusive lock (write lock) on the event counter file before reading or writing to it.
- If hepevt.py is unable to obtain an exclusive lock on the event counter file, it immediately returns http error 503 (server unavailable).
  - It is up to the client (i.e. TextFileGen module) to back off and retry.
- Argument “stream” can be used to give an event counter file a unique name.
- The username is incorporated into the event counter file name, if known (i.e. if using SSO authentication or invoked with argument “user”).

Hepevt.py Error Codes

- Hepevt.py may return the following http statuses.
  - 200 – success.
  - 400 – Event seek error (including end of file).
  - 404 – No hepevt file specified, or specified hepevt file not found.
  - 503 – Server (temporarily) unavailable.
- Other error statuses (mainly server errors 5xx) can be generated by the web server itself (as opposed to by script hepevt.py).

TextFileGen Module

- The TextFileGen module requires updates to work with server.
- The basic requirements are as follows.
  - Read hepevt events from http(s) server.
  - Retry server errors without crashing batch job or server.
  - Deal with security and authentication issues (optional).

TextFileGen http Client

- TextFileGen has been modified to read from either a file or a url.
- Read from url using libcurl C API.
  - Libcurl is a standard linux system component.

TextFileGen Error Handling and Retry Logic

- Hepevt.py may intentionally return status 503 (service unavailable).
- Other 500-series status codes (server errors) may be returned by by the server (i.e. without hepevt.py ever being invoked).
- Updated TextFileGen includes an indefinite retry loop.
  - All 500-series status codes (server error) are retriable.
  - All 300-series status (redirection) are handled internally by libcurl.
  - Any other non-success status codes (anything except status code 200) are treated as a fatal errors, and TextFileGen will throw an exception. Main possibilities:
    - 400 – Event seek error.
    - 404 – File seek error.
  - In case of retry, TextFileGen waits an increasing time delay between retries.
    - Similar like samweb, except delays are longer..
  - Maximum retry time set by fcl parameter.

TextFileGen FCL Parameters

```
*======================================================================
*
* FCL parameters.
*
* InputFileName - Name of hepevt input file (no default).
* Offset        - Number of events to skip (for file input, default 0).
* InputURL      - Server url (no default).
* Timeout       - Maximum server cumulative timeout (seconds, default 7200, 0=none).
* MoveY         - Propagate particles to y-plane (default don't propagate).
* UseSSOAuth    - Use SSO Auth (certificat/proxy based, default false).
* Cert          - Certificate file (defaults: $X509_USER_CERT, $X509_USER_PROXY,
*                 /tmp/x509up_u<uid>
* Key           - Private key file (defaults: $X509_USER_KEY, $X509_USER_PROXY,
*                 /tmp/x509up_u<uid>
* CertType      - Type of certificate file (default: libcurl decides).
* KeyType       - Type of key file (default: libcurl decides).
* KeyPasswd     - Key file password (default: none).
*
*======================================================================
```

Security and Authentication

- MicroBooNE's web server is public and accessible from anywhere, including, in particular, on site and off site grid nodes.
- Nevertheless, there are some reasons why it may be advantageous to consider adding security (SSO authentication).
  - Prevent interference from random people on the internet (probably not a major concern).
  - Fermilab could decide at any time to impose an SSO requirement on MicroBooNE's web server or other web servers, or SSO already required.
    - Fermilab allows public servers on central web hosting, but normally there is a review requirement.
  - SSO authentication allows the server to discover the authentiated user name associated with the client. Hepevt.py incoporates the user name into event counter file name, if it can discover it, or if invoked with argument “user.”

Security and Authentication

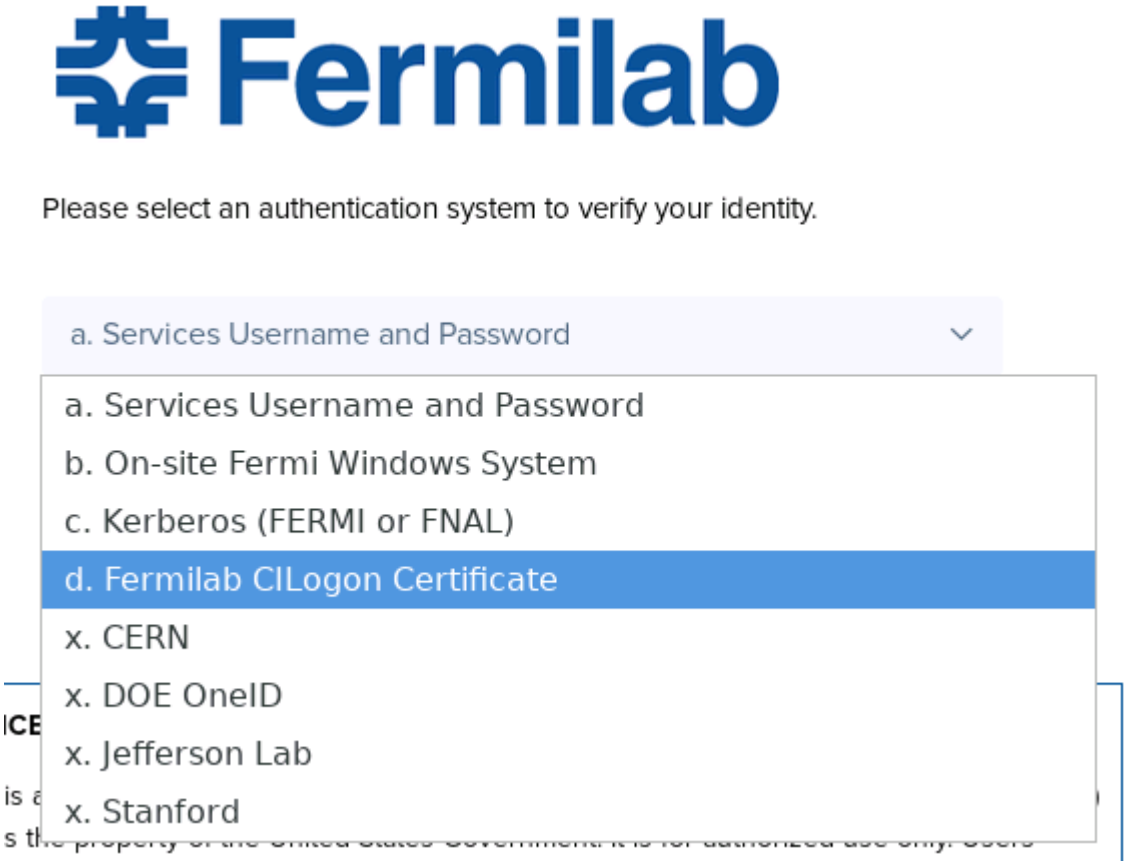
- For these reasons, I tried to teach TextFileGen to work with SSO authentication.
  - I actually installed two copies of hepevt.py in the MicroBooNE web site, one public and one protected by SSO.
    - Public server: https://microboone-exp.fnal.gov/cgi-bin/hepevt.py
    - Secure server: https://microboone-exp.fnal.gov/cgi-bin/hepevt/hepevt.py
  - I was only partially successful.
    - TextFileGen can handle the authentication handshake.
    - According to my tests, the MicroBooNE web server does not recognize grid proxies.

SSO Options

- When you access a Fermilab SSO-protected web page, you are presented with several options.

SSO Options

- But some web sites have extra options.



TextFileGen Using CiLogon

- CiLogon certificates can be obtained in the following ways.
  - kx509 command (cigetcert command).
  - Download directly from cilogon.org web site.
- Libcurl has options for dealing with certificates and keys.
  - TextFileGen tested working with either of the above types of CiLogon certificates.
- TextFileGen can be used interactively just by setting fcl parameter "UseSSOAuth=true" and typing "kx509."
  - Other fcl parameters can be left as default values.

TextFileGen Using Grid Proxies

- Unfortunately grid proxies are not cilogon certificates, and they don't work for standard SSO-protected web pages.
  - Grid proxies can be obtained in the following ways:
    - voms-proxy-init command.
    - SCD managed proxies.
    - Proxies sent to batch jobs by jobsub.
  - Typical error:

```
%MSG-e TextFileGen:  TextFileGen:textfilegen@BeginModule  24-May-2022 16:13:27
CDT run: 1 subRun: 0 event: 1
Curl returned status 35 from url https://microboone-exp.fnal.gov/cgi-
bin/hepevt/hepevt.py?file=HEPevents.txt
SSL connect error
error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
%MSG
```

SSO in Batch Jobs

- For the reasons on the previous slides, SSO-protected web pages are not (yet) accessible seamlessly in batch jobs.
- Options.
  - Stick with public web servers.
    - Jump through Fermilab hoops as necessary.
  - Teach server to recognize grid proxies.
    - I believe this is a server configuration issue (central web hosting).
  - Copy CiLogon certificates (or other recognized certificate type) to batch jobs.
    - This would at least be technically feasible, but:
    - Might not be compatible with Fermilab security policies.
      - CiLogon certificates are more powerful than grid proxies (kind of like having a copy of your Fermilab Services password in a file).

Performance and Scalability

- TextFileGen tested works interactively using public or secure (SSO-protected) web servers.
- TextFileGen tested in batch jobs using public web server.
  - I tested with cluster of up to 900 batch jobs.
    - Successfully triggered server errors and retry logic.
      - See log file next page.
    - 100% job success.
    - No noticeable lag when accessing MicroBooNE web pages interactively while batch jobs were running.

Example Log File

```
Begin processing the 1st record. run: 1 subRun: 197 event: 1 at 17-May-2022 05:44:36 UTC
TextFileGen: server unavailable, wait 10 seconds.
TextFileGen: server unavailable, wait 20 seconds.
17-May-2022 05:45:08 UTC  Opened output file with pattern "textfilegen.root"
%MSG-w FastCloning:  PostProcessEvent 17-May-2022 05:45:08 UTC  run: 1 subRun: 197 event: 1
Fast cloning deactivated for this input file due to empty event tree and/or event limits.
%MSG
Begin processing the 2nd record. run: 1 subRun: 197 event: 2 at 17-May-2022 05:45:09 UTC
TextFileGen: server unavailable, wait 10 seconds.
TextFileGen: server unavailable, wait 20 seconds.
Begin processing the 3rd record. run: 1 subRun: 197 event: 3 at 17-May-2022 05:45:41 UTC
Begin processing the 4th record. run: 1 subRun: 197 event: 4 at 17-May-2022 05:45:41 UTC
TextFileGen: server unavailable, wait 10 seconds.
TextFileGen: server unavailable, wait 20 seconds.
TextFileGen: server unavailable, wait 40 seconds.
TextFileGen: server unavailable, wait 80 seconds.
TextFileGen: server unavailable, wait 120 seconds.
TextFileGen: server unavailable, wait 120 seconds.
Begin processing the 5th record. run: 1 subRun: 197 event: 5 at 17-May-2022 05:52:25 UTC
17-May-2022 05:52:27 UTC  Closed output file "textfilegen.root"
```

TextFileGen Status

- I updated TextFileGen on the develop branch of my personal github fork of larsim (github user name hgreenlee).
  - Also added cgi server script in larsim/scripts/hepevt.py.
- Pull request made (larsim#95).
  - Includes updates to TextFileGen and hepevt.py cgi script.

Summary

- HepEvt server cgi script hepevt.py.
- TextFileGen updated to work with server.
- SSO authentication only partially solved.
- Pull request made.