Mu2e-doc-45719-v3

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Mu2e Framework Requests

Rob Kutschke

Fermilab Frameworks Workshop

June 5, 2023

Thank You

- Mu2e would not be where it is today effort put in by the art support team over the years to provide the *art* suite and related software.
- We appreciate the timely response to many issues over the years.

List of Issues/Requests

- Interruptable Modules and Debug Output Stream
- Extensions to fcl
- Deep Copy Filtering
- Object Extension
- TimeTracker Reports Wall Clock not CPU
- New Seed Type for RandomNumberGenerator Service
- Support for Seeding Using art::EventID
- Pywrap – Others might Find this Useful
- If There is Enough Time:
  - Sparse Vector Improvement
  - What Triggers ROOT Autoparse at Run-time?
  - Location of Temporary SQLite DB
  - Miscellaneous

Interruptable Modules and Debug Output Stream

- Mu2e trigger
  - time budget is ~5 ms /event
  - ~800 *art* processes running in parallel on 40 servers
- Suppose one module is running for too long (say 50 ms)?
- We would like to:
  - Interrupt the module
  - Skip processing the rest of the event
  - Write the event including data products available so far to a debug output stream.
  - Continue with the next event
- This will be very useful when commissioning with beam.
  - Curent schedule: summer 2026

Extensions to fcl

- We have many complex fcl files
  - Deep include hierarchies for single points of maintenance
- Development and maintenance often requires coordinated changes in multiple places
- Current solution is:
  - Prologs and epilogs
  - Scripts to write fcl files
  - Development and maintenance is tedious and error prone
- We request support for the following in fcl:
  - Conditionals: on-spiil vs off-spill configuration
  - Simple arithmetic with variables
  - See next page for examples

Extensions to fcl - Examples

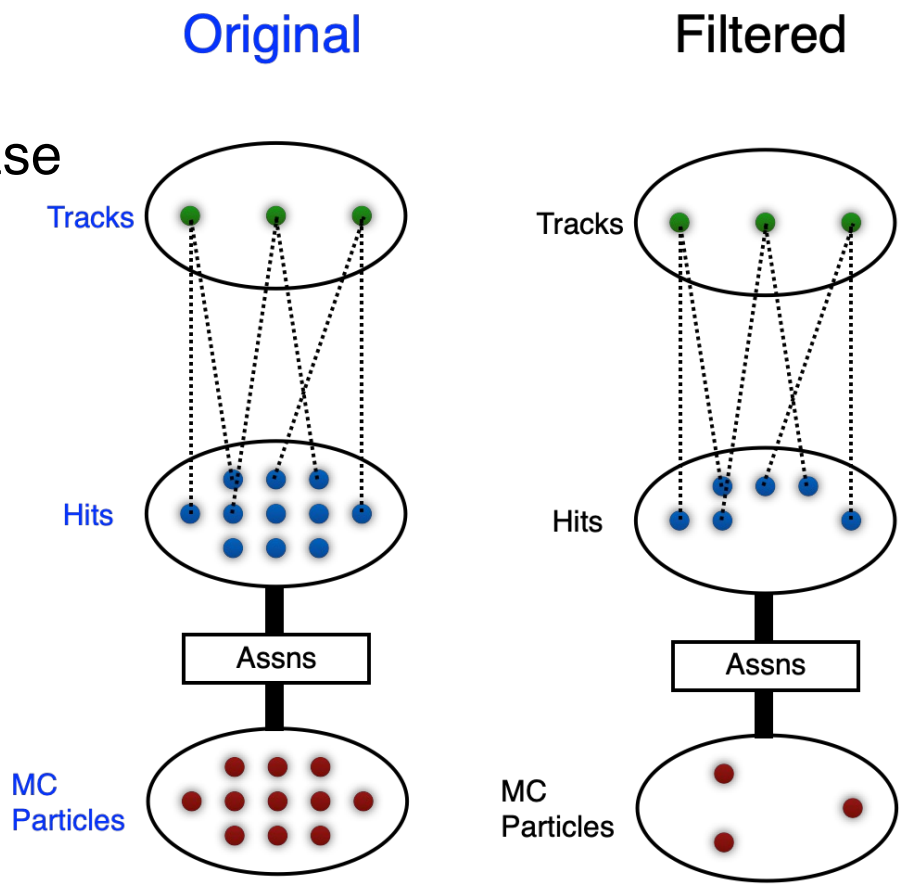
- Conditionals:
  - On-spill vs Off-spill
    - Event duration different
    - Touches many modules
- Simple arithmetic with variables

```
global_daq_start = 500; // All times inns
trk_daq_start    = global_daq_start -  50;
cal_daq_start    = global_daq_start -  10;
crv_daq_start    = global_daq_start - 100;
```

- Both cases were encountered during R&D
  - Often we need to make progress without waiting for infrastructure.

Deep Copy Filtering

- Assns and Ptrs create links between objects
  - Built up through simulation and reconstruction sequences
- Collection filtering is a common use case
  - Allows significant payload reduction
- Filtering linked objects requires deep copies of upstream and downstream objects and collections, rebuilding all links, …
- Mu2e has developed custom code to create deep copies
  - Intricate and error-prone
- Could art provide tools for deep copy filtering?
  - Based on a selector or selection?
  - Bookkeeping and validation tools?



Object Extension

- Some workflows involve extending existing event objects
  - ie extending a track fit to a vertex position
- Because event objects are const, extension requires a deep copy
  - Requires duplicate processing, object content
  - Requires updating any existing references to this object
  - Disruptive to downstream workflows
- Could art event support ‘object extension’?
  - Existing data content only appended to
  - Limited form of object modification
  - For Mu2e, in production we would only persist the extended object, not the original.

TimeTracker Reports Wall Clock not CPU

- art time tracker is reporting wall clock time
  - https://github.com/art-framework-suite/art/blob/4faa87c3f382a21517944281db5ff6d681c8dca6/art/Framework/Services/Optional/TimeTracker_service.cc#L70
  - uses std::steady_clock
- Ok-ish on lightly loaded machines for CPU intensive work
  - Horribly wrong on a heavily loaded machine
- We request:
  - An option to report CPU time

New Seed type for RandomNumberGenerator Service

- HepJamesRandom (art default; former CLHEP default)
  - Seed is a single long
- MixMaxRng (CLHEP default and Mu2e standard)
  - Supports seeding with a single long
  - Can also be seeded with an array of 3 longs
- Request additional overloads to createEngine to support a seed of type an array of 3 longs.

Support for Seeding Using art::EventID

- In G4MT each thread has its own RNG instance
  - To support repeatability for debugging, Mu2e reseeds each RNG instance at the start of each event using:
    - A salt and the `art::EventID`.
    - Using MixMaxRng seed of 3 longs.
  - Salt is a parameter of the module; the same for all threads.
    - Supports reprocessing the same input event with either repeatability or uniqueness as needed.
- Request that art support this feature
  - We think it will be useful to others
  - We may wish to use this in other modules

Pywrap – Others might Find this Useful

- For python analysis, we had requests to
  - Access C++ enums used in data products
    - enum of MC particle fate
  - Run utility executables
    - dbTool for accessing calibrations
  - Run C++ utility functions inside python
  - No support for modules, services
- What we implemented
  - Build and package swig 4.0.2 in a UPS product
  - Add swig config to the Offline repos in pywrap.i files
  - Add swig commands to the scons build process
  - Write python classes and hooks to the build area, add to python path
- Result
  - It satisfies the request and, so far, has been robust
  - building swig configs to deal with advanced c++ features can be tricky

If There is Enough Time …

Sparse Vector Improvement

- Current implementation of a sparse vector
  - `cetlib/map_vector.h`
  - `using value_type = std::pair<key_type, mapped_type>;`
  - `using impl_type = std::vector<value_type>;`
- Request a new class template for a sparse vector in which key and value are maintained in separate vector's
  - Less cache thrashing for key lookup when mapped_type has a large memory footprint.

What Triggers ROOT Autoparse at Run-time?

- root sometimes needs to parse header files at run time
  - Solved by correctly maining ROOT_INCLUDE_PATH at run time
- It seems like there should be a way to build dictionaries completely at build time.
  - If there is a way, how do we do it?
  - See backup slides for the commands we use to build dictionaries.
  - The example on which we drilled down a few years ago involved a data product that contained an std::array<T,N>.
  - Email thread between Ray and Philippe:
    - eg August 4, 2021 5:33 PM

Location of Temporary SQLite DB

- https://github.com/art-framework-suite/art/issues/132
- Enabled TimeTracker on a job running 10M events
- Default location for temp SQLite db was /var/tmp
  - Which filled up on a long job
- Fixed using:
  - `--timing-db /path/to/filename.db`
- An open question:
  - What should the default location for the db?
    - Should it default to current working directory?
  - Does the community have a consensus?

Miscellaneous

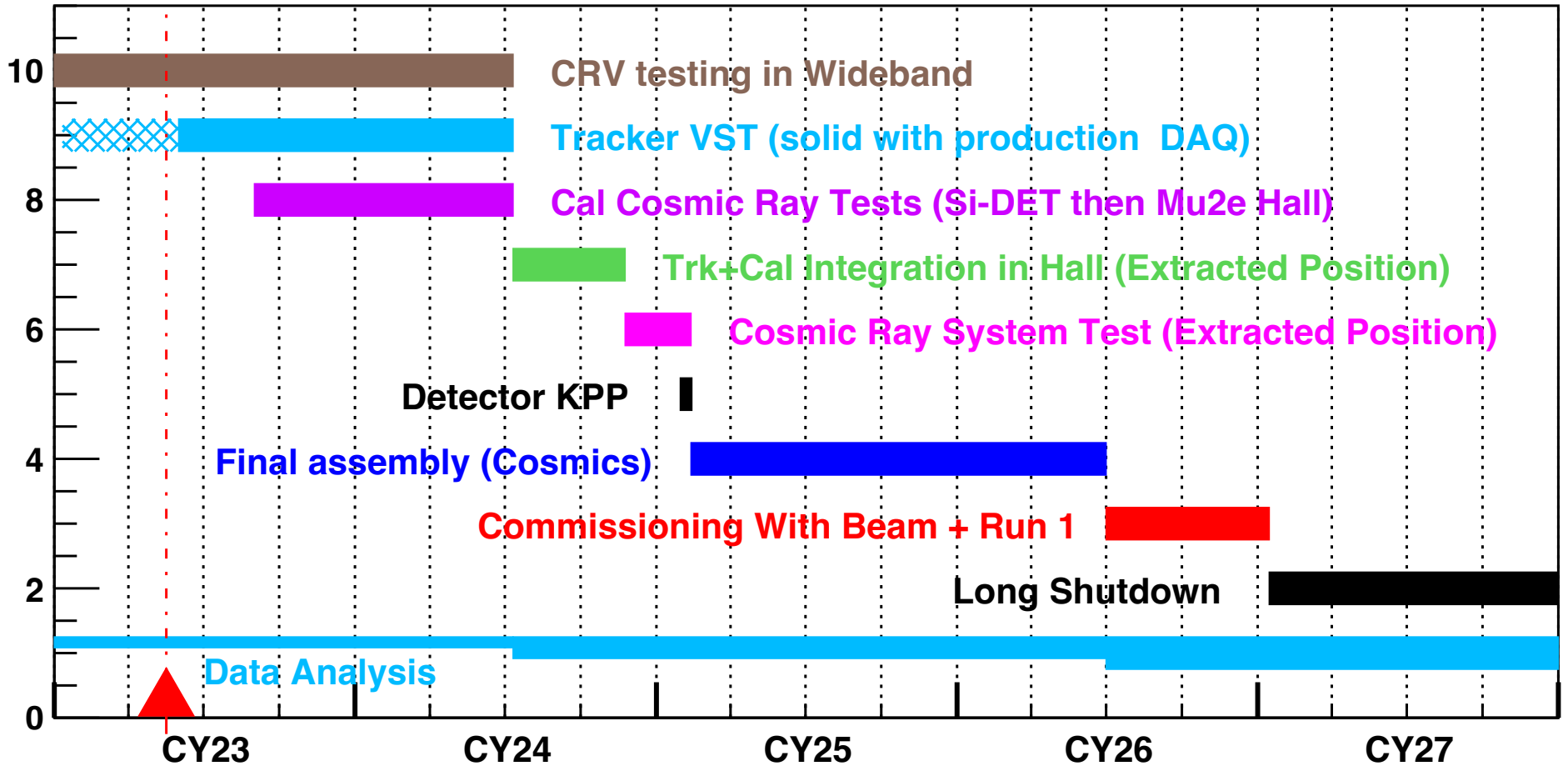
- What is the timetable for ifdh_art to support RUCIO/Metacat/DataDispatcher ?
- Mu2e builds our own art main program
  - To provide a banner with version numbers, build location, build time, etc
  - Could art support a plugin we can use to implement our banner?
  - Others might find this useful

Conclusions

- Again, thanks for the support over the years.
- We hope that other experiments will also find many of our requests and ideas useful for them.

Backup Slides

Mu2e Schedule: Informs Computing Milestones



Review Date

Start of high volume data taking

Fermilab

Example command to build a dictionary

```
genreflex Offline/DataProducts/src/classes.h \
   -s Offline/DataProducts/src/classes_def.xml \
  -I.  \ # More -I options elided.
  -l build/sl7-prof-e20-p041/Offline/lib/libmu2e_DataProducts_dict.so \
  -o build/sl7-prof-e20-p041/Offline/tmp/DataProducts/src/dict/mu2e_DataProducts_dict.cpp \
  --fail_on_warnings \
  --rootmap-lib=build/sl7-prof-e20-p041/Offline/lib/libmu2e_DataProducts_dict.so \
  --rootmap=build/sl7-prof-e20-p041/Offline/lib/libmu2e_DataProducts_dict.rootmap -DNDEBUG -v


g++ -o build/sl7-prof-e20-p041/Offline/tmp/DataProducts/src/dict/mu2e_DataProducts_dict.os \
    -c -std=c++17 -Wall -Wno-unused-local-typedefs -g -Werror \
    -gdwarf-2 -Werror=return-type  -Winit-self -Woverloaded-virtual \
    -O3 -fno-omit-frame-pointer -fPIC -DNDEBUG \
    -I. \ # More -I options elided.
    build/sl7-prof-e20-p041/Offline/tmp/DataProducts/src/dict/mu2e_DataProducts_dict.cpp

g++ -o build/sl7-prof-e20-p041/Offline/lib/libmu2e_DataProducts_dict.so \
   -Wl,--no-undefined -Wl,--as-needed -shared \
   build/sl7-prof-e20-p041/Offline/tmp/DataProducts/src/dict/mu2e_DataProducts_dict.os
    \
    -Lbuild/sl7-prof-e20-p041/Offline/lib         \ # more -L options elided
    -lmu2e_DataProducts -lart_Persistency_Common    # more -l options elided
```

Giani's Time Tracker Issue

- I need input from Giani
- It is something about needed to define extra paths in order to get understandable timing results

```
t1   : [ p1, f1, p2, f2, p3, f3]
t1_0 : [ p1, f1, p2, f2, p3]
t1_2 : [ p1, f1, p2]
```