

Lazy Database Services

Larsoft Coordination Meeting
Nov. 20, 2018

H. Greenlee

Outline

- Review of proposed changes to IOV database services.
- Current status.
- Validation.

Provider Updates

- New functions and data members.
 - Add data member `fCurrentTimeStamp` – Time stamp of cached data.
 - Add data member `fEventTimeSTamp` – Time stamp of most recent event.
 - Add public function `UpdateTimeStamp(timestamp)`
 - Updates `fEventTimeStamp`.
 - Add private const `DBUpdate(timestamp)` function.
 - Does actual update and updates `fCurrentTimeStamp` (if necessary).

Provider Updates (cont.)

- Changes to existing functions and data members.
 - Existing function `Update(timestamp)` (public, non-const) updates `fEventTimeStamp` and calls `DBUpdate(timestamp)` (private, const).
 - Make all data members modified by `DBUpdate(timestamp)` mutable.
 - Call `DBUpdate(fEventTimeStamp)` function from all other functions (accessors) that depend on mutable data members.

Proposed Changes to Services

- Modify `PreProcessEvent` callback to call provider function `UpdateTimeStamp(timestamp)` instead of `Update(timestamp)`.

Commentary on Proposed Changes

- Nonbreaking interface change.
 - Only visible change is addition of new provider function `UpdateTimeStamp`.
 - Functions `UpdateTimeStamp` and `Update` do the same thing. The former is lazy about database access, but the latter is not.
- Art service `PreProcessEvent` callback becomes inexpensive because it can never trigger a database read.
 - Database access triggered by accessor function call.
- Existing callers of provider function `Update(timestamp)` might benefit by calling `UpdateTimeStamp(timestamp)` instead, but they won't break if they don't make this change.
 - Only possible if they ignore return value of `Update(timestamp)`.

Current Status

- Since my previous in the previous larsoft coordination meeting, there have been more instances of database overloading, possibly triggered by MicroBooNE's recent optical filtering campaigns.
- Updates to database services and providers have been made following the plan presented two weeks ago and repeated in this talk. Branches are ready to merge.
 - `larevt – feature/greenlee_lazy_db_develop`
 - (Branch `feature/greenlee_lazy_db` is for MicroBooNE's `mcc8` branch).

Updated Larsoft Services and Providers

- In `larevt/larevt/CalibrationDBI`.
 - `SIOVChannelStatusService / SIOVChannelStatusProvider`.
 - `SIOVDetPedestalService / DetPedestalRetrievalAlg`.
 - `SIOVElectronicsCalibService / SIOVElectronicsCalibProvider`.
 - `SIOVPmtGainService / SIOVPmtGainProvider`.

Services and Providers

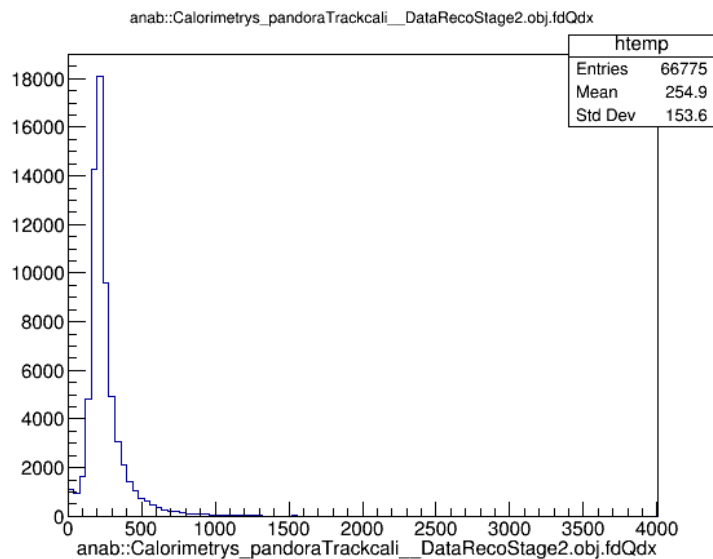
(Owned by uboone suite)

- In [uboonecode/uboone/Database](#) or [ubevt/ubevt/Database](#).
 - [UbooneChannelStatusService](#) (service only).
 - [UbooneDetPedestalService](#) (service only).
 - [UbooneElectronLifetimeService](#) / [UbooneElectronLifetimeProvider](#).
 - [UbooneElectronicsCalibService](#) / [UbooneElectronicsCalibProvider](#).
 - [UboonePmtGainService](#) / [UboonePmtGainProvider](#).
 - [UbooneTPCEnergyCalibService](#) / [UbooneTPCEnergyCalibProvider](#).

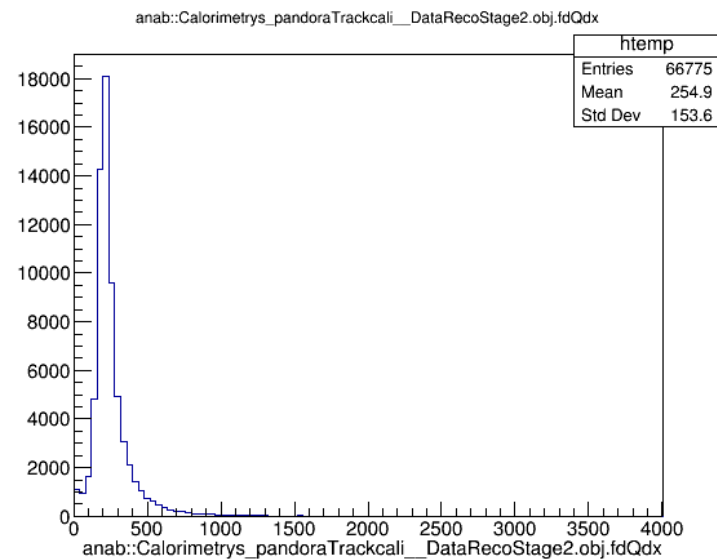
Validation Tests

- Ran five data events through standard reconstruction, with and without lazy database updates. Verified that uncompressed size of each branch is unchanged (exact number of bytes).
- Verified that calorimetry information is unchanged.

Original dQ/dx



Lazy dQ/dx



Branches

- Larsoft.
 - Develop branch.
 - Larevt – feature/greenlee_lazy_db_develop
 - Merge request for next integration release.
 - MCC8 branch (v06_26_01_01_br).
 - Larevt – feature/greenlee_lazy_db (already merged).
- Uboone suite.
 - Develop branch.
 - Ubevt – feature/greenlee_lazy_db
 - Ubevt – feature/greenlee_lazy_db_no_larsoft (if no larevt merge).
 - MCC8 branch (v06_26_01_br).
 - Ubooncode – feature/greenlee_lazy_db