protoDUNE/SP Raw Data Access

Brett Viren

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

DAQ Collab Week Prep
19 Jan 2017

Outline

Data Access Requirements

Raw Data Clients

Data Access Requirements

This is mostly a summary of recent email threads.

file names universally unique, human-oriented semantic labels

- eg: `pdsp-raw-<runtype>-<runnum>-<subrun>.root`
- bonus if lexical sorting gives time-ordering!

file size ~5 GB

file format ROOT, likely uncompressed

metadata file JSON formatted file holding summary values

- holds run, time/trigger range, detector config name, etc. details to be determined.
- one JSON produced along side each ROOT file
- used for registering data file into file catalog (SAM)

High-level File Schema

fragment	one array for each from each board reader
grouping	subdetector fragments grouped together
fragment array	composed of artDAQ **header** + **payload**
header	common artDAQ packing schema across fragments
payload	BR-specific packing (more on this)
event	collection of fragments from a common trigger
file	collect N events to reach $\sim$5 GB/file

Event Definition and Identifiers

ident each event uniquely defined by run and an event identifier

contiguous events span non-overlapping times

span event covers ~5 ms around trigger

contents TPC waveforms, PDS (self trigger + few us waveform), trigger and timing, possibly muon tagger data and some beam info.

- BI is still out-of-band?

Raw Data Access Library - "raw DAL"

dependencies access data with "minimal" set of software dependencies (eg, only ROOT?)

payload DAQ-provided, shared code to pack and unpack **fragment payload** arrays.

header artDAQ packs/unpacks **header arrays**, would like to have this code available also outside of artDAQ.

versioning artDAQ header and payload arrays place a **packing schema version identifier** in a "well known" location in their arrays and update it for any change in the packing schema.

release new version of raw DAL is tested, released and distributed before any new packing schema is deployed in production DAQ.

evolution new versions of raw DAL must read old versions of data.

TPC common interface to access RCE and FELIX APA data

granularity unpack one APA w/out need to read/unpack others, unpack SSP w/out need to unpack APAs, etc.

Numbering Conventions

- Want to document existing numbering conventions in use right now by different groups.
- Try to define an **common convention** that is least disruptive to existing use.
- Provide **mappings** between domain-specific and common convention in both documentation and code.
- Ideally, DAL uses common convention o.w. provide mapping from DAQ-domain to common.
- I promised to organize this but it's not done, Mea culpa!

Some Work Needed Toward Data Access

1. Numbering convention organization
2. Initial packing schema and raw DAL implementation
3. Test code to write example files with raw DAL
4. Test reading into various clients
5. ...

Raw Data Clients

Before files, in artDAQ (both packing and unpacking)

- artDAQ DataLogger
- FELIX APA compressor
- Online Monitor

After files (unpacking or file-level)

- Fermilab-File Transfer System
- Detector Debugging Experts
- Prompt Processing
- Data Reduction
- Production Processing

Some thoughts on each client follow if there is time....

artDAQ DataLogger

- sees full events "instantly"
- writes events to $\sim$ 5 GB ROOT file on online buffer disk
- artDAQ header packing, treats payload as opaque (?)
- contract with F-FTS:
  - write SAM metadata JSON file,
  - provide "file-ready" notification (eg, via a `mv` to a watched directory).

FELIX Compressor

- Uncompressed FELIX APA data $\rightarrow\sim 2\times$ more raw data
- FELIX compression in artDAQ
  - Yes, please!
- **needs raw DAL** for unpacking and packing

Online Monitor

- sees either fragments or fully built events "instantly"
- inputs artDAQ stream, no data files
- will write summary/visualizing files
- **needs raw DAL** for unpacking
- potentially sees uncompressed FELIX APA data

Fermilab File Transfer System

- **most important client** of data files
- reads all artDAQ files from online buffer storage
- treats data file contents as opaque
- watches for "file-ready" notification
- ingests SAM metadata JSON file
- transfer to EOS, CASTOR and beyond
- **deletes** its input files (configurable conditions)

Detector Debugger Experts

- Needs **immediate access** (∼minutes) to small number of particular raw files.
- Needs fast and lightweight access to data
- **needs raw DAL** for unpacking
- may use LArSoft, plain art or potentially stand-alone code (plain ROOT, dump out to allow Python or Mathematica)

Prompt Processing

- Inputs 1-10% of files from EOS or online buffer via XRootd (buffer access needs discussion/assurances)
- Wants low latency ($<$1min) file access and robust against 20Gbps linke outage.
- Jobs must unpack artDAQ headers and payload
- **needs raw DAL** for unpacking
- may be LArSoft, plain art or potentially stand-alone code
- writes summary and visualization files for later display by p3s web server

Data Reduction

- Reduce data by 400x while keeping all useful signal after noise subtraction and signal processing (2.5 PB $\rightarrow$ 6 TB)
- Run it to keep up with data taking: 45s/trigger/APA.
- Still keep full raw data, can rerun reduction later.
- **needs raw DAL** for unpacking **reduced DAL** for packing
- may be LArSoft, plain art or potentially stand-alone code
- writes reduced data, probably in same format as artDAQ but different packing. Uses F-FTS to archive/distribute.
- nominally "shunts" non wire data fragments as-is, to output but could do something more useful.
- would be a good place to merge in Beam Info.

Production Processing

- Runs on reduced data, if we can run in "keep up" mode
- LArSoft based code with input module
- needs **raw DAL** or **reduced DAL** for unpacking
- Likely uses SAM to select input file sets.