Event processing frameworks

Alden Fan (SLAC/LZ)

16 November 2021

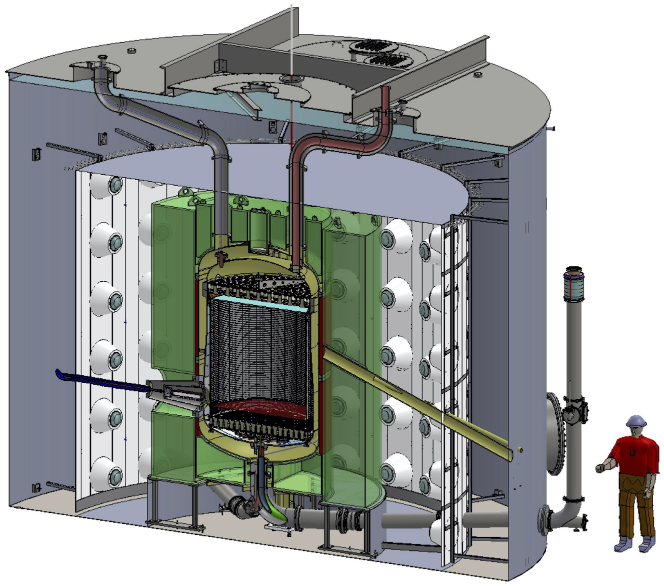
Software and Computing for Small HEP experiments

Outline

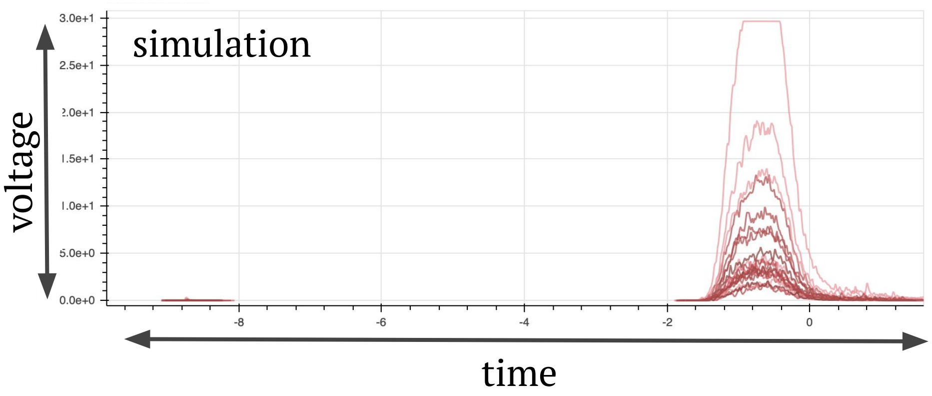
- Event processing in LZ
- Event processing for small scale experiments in general
- Some comments on long term

* The ideas expressed here are my own and do not necessarily reflect those of the LZ collaboration.

LZ data



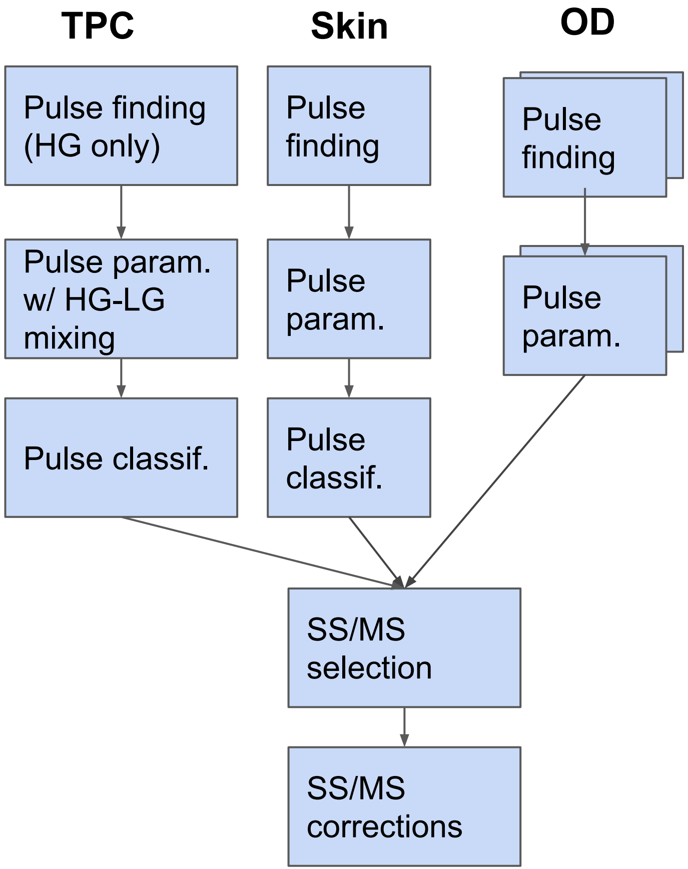
DAQ data are PMT traces



- DAQ data:
  - ~1400 channels digitized at 100 MS/s, zero-suppressed
  - ~40 MB/s
  - Self-triggered
- Run control data
- Environmental data:
  - 1000s of sensors (temperature, pressure, voltage, etc)
- Calibrations constants

LZ reconstruction

- **LZap** = LZ analysis package (poorly named)
- Built on **Gaudi**
- Deployed via cvmfs
- Three ~independent lower level chains that merge into a single higher level chain
- Peak finding, classification, position reconstruction (fitting)
- Retrieve environmental data, calibration data, run control data for each event
- Output: **ROOT** TTree of (very) jagged arrays… it's vectors all the way down

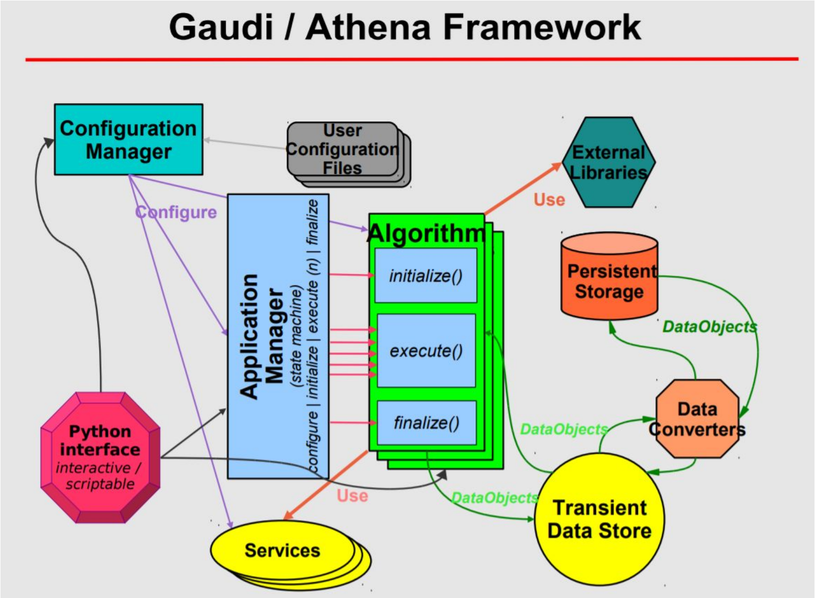


LZ analysis

- **ALPACA** = tortured acronym (Analysis Lz PACkAge)
- Thin wrapper around TTreeReader. Essentially, **a lightweight framework**.
- **Core code** to manage the event loop + user-written **analysis modules**.
  - **1 analysis = 1 module**
  - Helper scripts to quickly generate new modules
- Includes multiple services: histogramming, skimming, ML integration, more
- Anyone can easily run anyone else's code
- Features, especially cut definitions, feed back into common codebase
- Extensive documentation.
  - New features merged to core code only when documentation is updated.
- **Widely adopted**: >60 users (~250 person experiment) and >100 modules

Frameworks

- "Small experiment" ≠ "small computing problem"
- Small experiments need nearly every feature of full-fledged framework:
  - Configuration management
  - I/O: Transient and persistent data models
  - Services:
    - Conditions & calibrations
    - Messaging
    - Provenance tracking(?)
    - etc
  - Algorithms: Sequencing, filtering
  - Multithreading
  - External libraries
- But not always a perfect fit…



Unique challenges

Small experiments can bring unique challenges

- For self-triggered data, precise timestamps and time-ordering of events is critical (DAQ)
- For liquid noble detectors, there are detector effects that often span multiple consecutive triggers → require sequential processing of events, ability to look at more than one event a time
  - Challenge for adopting existing tools from collider experiments
  - Especially columnar data processing
- …

Provenance tracking

- Frameworks provide the ability to capture the full processing history of every event
- For LZ and DarkSide, this feature is not much utilized:
  - Running reconstruction requires data and compute resources too large to do anywhere except in managed productions
  - → History is captured by software releases and metadata tracking

Data slimming/skimming

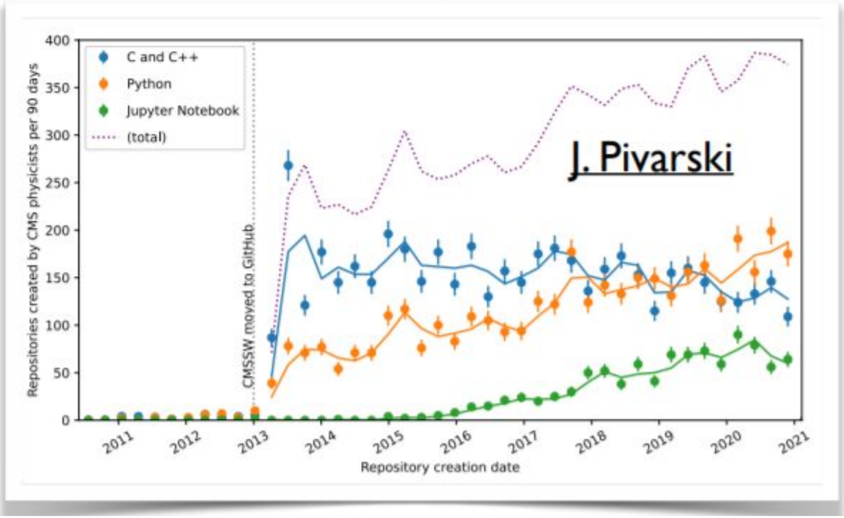
- (My very biased view)
- Liquid xenon detectors have a long history of surprises - unexpected detector effects.
- Systematics from reconstruction are complicated.
- Provide as much information downstream as possible, i.e. no slimming or skimming, especially in early days of an experiment
- → Need a downstream analysis model that supports this:
  - I/O speed vs. ease of access (particularly in ROOT/PyROOT)
  - Method to deploy common/core cuts, until centralized skimmed/slimmed data can be produced
    - Conflicts with BYO-tool
- LZ (mostly) achieved this with ALPACA
  - ALPACA can provide official cut definitions; analyzers choose which ones to apply
  - ALPACA can skim data ad hoc

Deployment and maintenance

- How do new experiments build up their reconstruction package?
  - Deployment of a new framework requires expert knowledge
  - Fermilab experiments using *art* benefit from Fermilab support
  - Other experiments must rely on prior or outside expertise
  - LZ as an example: no significant computing ties to Fermilab or CERN; worked hard to secure support for usage of Gaudi
    - Missing support for xrootd out of US data center
- Who will maintain the framework throughout the lifetime of an experiment?

Algorithm development

- Python is increasingly the preferred language in HEP and data science
- Reconstruction frameworks in C++
- For small experiments: reconstruction algorithms largely developed by students/postdocs → loss of potential contributors to reconstruction algorithms.
- Trending to diverge more in the next 10 years.



CHEP2021

New repositories in CMS vs. time
Trend applies to small experiments

Conclusion

- "Small experiment" ≠ "small data volume" or "small computing problem"
- Community-supported event processing frameworks are a good fit for small experiments
- But they require expert support that is sometimes (or often) lacking in small experiments

Thanks for many useful discussions with M.E. Monzani, Steffen Luitz, Simon Patton, Gianluca Petrillo, Kazuhiro Terao