DUNE Requirements for LArSoft

Tom Junk, Amir Farbin

Context

- There seems to be general anxiety regarding the readiness and effectiveness of automatic reconstruction for LArTPC.
  - Not necessarily accounting for the incredible progress in the past year…
  - Easy topic to point to during reviews…
- LArSoft Requirements workshop (Oct 19-20) https://indico.fnal.gov/conferenceDisplay.py?confId=10394
  - We should differentiate between:
    - **Technical requirements:** e.g. OS support, Memory, etc…
    - **Physics requirement**: e.g. tracking or PID efficiency, energy resolution. Disambiguation?
      - My opinion: Not clear if these are really software or detector requirements. SW should do as well as the detector allows.
    - Note that LArSoft already meets many requirements. Nonetheless we need to identify them.
- While LArSoft team has provided core and technical elements, reconstruction algorithms have been developed inside experiments.
  - Need to understand the model of developing reconstruction within multiple experiments.
    - Who ensures that alg from experiment X is properly incorporated into LArSoft and used by other experiments?
    - Where does the experiment/LArSoft boundary lie? What do we expect from LArSoft?
    - How do we properly give credit where it's due? Essential to the health of the long term project.

Context/Scope…

- My perspective: Coming from the LHC I need to constantly renormalize my vision.
- Keep in mind: Data sizes and reconstruction rates are not the necessarily the appropriate metrics to assess the software performance.
    - Roughly speaking, DUNE data will be ~ LHC Run 1… we will not be hardware limited in 2025.
    - With zero-suppression, cosmic and ^{39}Ar rejection, DUNE beam spill data is few TB/year.
    - Caveats are:
        - protoDUNE and Near Detector will have much higher data rates.
        - We may adopt some ambitious Supernovae/low energy plan.

	Nominal	reduce beam intensity	DAQ group's #drifts	reduce drifts, <1 duty	Turn on ROOT compressio n, DC readout
drift time (ms)	2.25	2.25	2.25	2.25	2.25
spill time per rep (s)	9	9	9	9	9
beam rep (s)	45	45	45	45	45
particles per second	200	148	200	200	200
drifts readout	3	3	2.4	2.2	2.2
cosmic size (MB)	1	1	1	1	0.1
cosmic rate (/ms)	10	10	10	10	10
trig eff	1	1	1	1	1
det duty	1.35	**0.999**	**1.08**	**0.99**	0.99
muon/RO	67.5	67.5	54	49.5	49.5
RO size (MB)	67.5	67.5	54	49.5	4.95
inst rate (GB/s)	13.5	9.99	10.8	9.9	0.99
spill size (GB/spill)	121.5	89.91	97.2	89.1	8.91
avg rate (GB/s)	2.7	1.998	2.16	1.98	0.198
24h volume (TB)	233.28	172.6272	186.624	171.072	17.1072

- Analysis is really reconstruction: Unlike colliders, different physics will likely requires it's own fine-tuned reconstruction chain.
- Very different software development model that is distributed across many experiments.

Requirements

- Tom produced a first list of requirements…
- I'm presenting here my first attempt to organize and iterate. Needs a lot more input, thinking, work…
- Let's use this meeting to identify what we missed…
- Fundamental requirement:
  - Store, Simulate, Reconstruct, and Analyze Data from 35t, protoDUNE, Near Detector, and Far Detector all within same software framework.
- Where is LArSoft used? Online/offline…
  - ArtDAQ provides an online environment for DAQ and triggering(?)
    - Zero-Suppression probably done online,
  - But any higher-level triggering based on reconstruction is probably better suited for offline.
    - Running offline software online???

Data Requirements

- All data always readable: from first test beam to the end of the experiment plus post-experiment analysis time (estimated: 30 years for DUNE)…
    - *Schema evolution*: ability to evolve data products over time, yet still read old data. We should understand what ROOT supports.
    - *Transient/Persistent Separation*- Different representations of stored and in-memory data?
    - at some point we may need to review the underlying storage technology.
- Ability to fine-tune data content, for example:
    - raw::RawDigit entries only on a portion of channel, perhaps associated to specific tracks/clusters or with cosmics or ^{39}Ar removed.
    - Support multiple data tiers e.g. Raw, Reco (aka ESD), Analysis (aka AOD)…
        - We will likely need the ability to (excuse the horrible ATLAS nomen-culture here)
            - *Skim*: store subset of events
            - *Thin*: store subset of objects in events
            - *Slim*: store subset of info inside objects
            - *Augment*: add arbitrary higher-level data to events and data products
- Store Object *Association*: e.g. Reco<->Truth map.

Data Requirements

- Flexibility to handle different *data compression* techniques like Huffman coding, Zero suppression, and other future ideas.
- *Event-pick facility*- ability to identify and navigate to events based on small set of parameters.
- Ability to *split* (in time/region), *stitch*, *mix/overlay* Events
- Ability to *evolve* and *expand* the Event Data Model, example
    - Wire cell needs a data product capable of holding charge+point info and which is not tied to a track
    - New data objects, perhaps composed of others, e.g. electron, pion, neutrino, …
- Store *Truth* and associate Info at different levels (hit, tracks, electron, …)
- Data *instrumentation/monitoring*: Keep track (for optimization?) of event counts and i/o statistics -- file counts, compression, bytes in/out, time used, data accessed (?)
- Ability to *optimize* data storage for speed and size.

Simulation & Reco

- Simulate and reconstruct a wide range of detectors within same framework:
    - Single-phase, dual-phase, and Gaseous TPC's.
    - Multiple TPCs: be able to stitch events spanning multiple TPCs.
    - Non-TPC detectors: Near Detector, Auxiliary Detectors
- Associate across (sub-)detectors (and performance metrics)
    - TPC/Photon
    - External Auxiliary Detectors
- Additional Features:
    - Space-charge sim/reco
    - Photon detector flash sim/reco
    - Space, time, and charge resolutions
    - Disambiguation
    - T0 finding/matching across detectors.
- No detector fiducial cuts should be made during the reconstruction, i.e. hits, tracks, or other objects should not be required to be within the TPC volume in the reconstruction stage.

Geometry

- Flexible geometry -- new experiments and configurations must be easy to model.
  - Easily establish and integrate non-LArTPC detectors.
  - Currently some things are a little hard-coded like the maximum number of wire planes in drift ionizaiton electrons and the number of different electric fields
  - Allow for alignment shifts (as-built and calibrated with cosmic or other alignment techniques)

Event Display

- Need different Event Displays:
  - Fast 2D event display, features
    - TPC navigation: next/prev TPC navigation thumbnails of TPC's (for fast drill-down), automatic selection of most active TPC, views with the most activity
    - Zoom, adjustable color palette, …
    - Adjustable Noise cut?
  - 2D display w/ reconstructed objects
    - Display all levels of data: raw waveforms, intermediate "fixed" waveforms, deconvoluted/calibrated waveforms, hits, clusters, tracks, showers, vertices
  - Photon detector event display -- alongside TPC event display
  - 3D event display showing 3D space points and reconstructed objects
  - Scanning tools (recording user classification)

Infrastructure

- Documentation, tutorials, training, …
- Build tools: Speed, Portability (multiple OSs), Optimized and Debug versions (debuggers)
- Code repository: Public availability. Backed up. Reliable (i.e. not susceptible to long downtime due to hardware failure or misconfiguration)
- Debugging tools.
- Be able run on laptops, GRID, …
    - Code/release distribution
    - Access to databases
    - Access to data sources (e.g. xrootd)
- Book keeping and error reporting…

Databases

- We need implementations and/or interfaces for (not necessarily distinct)
  - Geometry? We'll eventually have survey and alignments.
  - Configuration- how we configured detector/trigger
  - Conditions- e.g. calibration parameters
  - Ambient- temperature, purity, …
  - Meta-data- data type, …

The Long view

- We are designing a experiment that will start in early/mid 2020s and run for decades… hardware/software will significantly evolve until then and during running.
    - Ideally we would be architecture independent- always be able to use newest tech
    - And optimizable for specific architectures.
        - Suggests perhaps decoupling high-level code from back-end.
- Significant increase of cores/CPU will affect us
    - *Multi-core CPUs*: we are quickly approaching 100's of cores/CPU. Running an instance of same job on each core, each processing it's own events won't work.
        - Not practical to have 4 GB/core…
        - Not enough bandwidth to memory…
        - *Task Parallelization*: e.g. lots of threads sharing same memory, but each processing a single event through an algorithm
    - *Many-core Co-processors (possibly within CPU dye)*: GPUs/MiCs, FPGA, ASICs
        - Optimized for high throughput not low latency.
        - Requires different Parallelization model (Data Parallel) where many events are simultaneously processed in single algorithm.

The Long view

- HEP won’t be able to rely on the embarrassingly parallel nature of our data processing…
    - We just many instance of same software processing different events.
    - DUNE computing requirements may minimal enough that we can not worry too much …
        - Caveats are Wirecell and efficient use of HPCs
- Concurrency (simultaneously processing many events) is a hot topic. 2 types
    - Many threads processing one different event each
    - Algorithms processing many events at once.
- The LHC experiments are confronting this issue. Current focus on Task Parallelism:
        - CMS already has multi-threaded ART.
        - ATLAS using plans to build on Gaudi-Hive for Run 3.
        - There are schemes to push some algs to co-processors… but not fundamental in the design.
    - Data Parallelism is hard and rapidly evolving… strategy is not yet clear.
- We may think about making sure that the fundamental design of LArSoft is not incompatible with threading.
- We should consider that we may have to build completely new framework on the time-scale of DUNE and HL-LHC (ie early 2020s) and rewrite everything in order to be able to take advantage of latest hardware.

Physics Requirements

- Ideally we physics groups should provide he performance required for each physics measurement.

- Again, it is clear that these are not strictly software requirements… not clear if they should be LArSoft requirements or DUNE Reco requirements.

- Basics:

    - Tracking efficiency > 95%

    - Vertex position resolution: 2.5 cm in all three dimensions (probably need better than this in order to have e-gamma separation topology performance)

    - Short-sub finding efficiency (10 hits or more, all views together): > 90%

    - e/gamma separation: 90% efficiency for electrons, 99% rejection of photons from pi0 decays using both dE/dx and topology

Physics Requirements

- Assumptions from the DUNE CDR Volume 2, Table 3.3
  - *Stopping Track*: energy resolution of 5%
  - *Showering or exiting* energy resolution of 30%
  - *Muon* detection threshold of 30 MeV, angular resolution of 1 degree
  - *Charged Pion* detection threshold 100 MeV, angular resolution of 1 degree
  - *Electron and photon* detection threshold: 30 MeV
    - EM shower energy resolution: 2% ± 15%/sqrt(E) where E is in GeV
    - EM shower angular resolution: 1 degree
    - EM energy scale uncertainty: <5%
- *Proton* detection threshold: 50 MeV
  - energy resolution: 10% for p<400 MeV, 5% ± 30%/sqrt(E) for p>400 MeV (E is in GeV)
  - angular resolution: 5 degrees
- *Neutron* detection threshold: 50 MeV (KE)
  - energy resolution: 40%/sqrt(E) where E is in GeV
  - angular resolution: 5 degrees
- Other particles (K, Lambda, Sigma, deuteron) detection threshold: 50 MeV (KE)
  - energy resolution: 5% ± 30%/sqrt(E)
  - angular resolution: 5 degrees