Benchmarking & Validation

Wouter Deconinck, for ePIC CompSW
February 08, 2023

Goals for this presentation

- What do we have now, and how do we use it?
- What are the reasons for benchmarking and validation?
- How can the SWG work with the DWG and PWG to provide validated and reproducible “standard candle” plots?

Current Benchmarks on eicweb

Goal: Validate input to detector and physics studies for ePIC

Three classes of benchmarks needed:

1. **detector benchmarks**: based on output of detector simulation (single configuration), e.g. used to check basic detector quantities and calculate sampling fractions, material budget plots, etc.
2. **reconstruction benchmarks**: based on output of detector simulation or digitization, intended to test reconstruction algorithms, e.g. determine tracking resolution and efficiency, etc.
3. **physics benchmarks**: based on output of reconstruction, intended to mimic reference physics analyses, e.g. DIS x and Q^2 resolutions, DVMP, etc.

Most benchmarks ‘kept warm’ by running on O(100) events for every commit to geometry and reconstruction framework, and run for O(10k) events on a weekly basis

Benchmarks as a source of best practice examples

During ATHENA and even since then, benchmarks turned into a source of analysis examples, written in multiple modern ways: RDataFrame and python-based.

There may be benefits if benchmarks are organized into individual repositories managed by DWG or PWG instead of one mono-repository with all benchmarks (this gives implicit notion of ownership to WG).

Benchmarks must make a pass/fail decision to be useful as tests, and provide a numerical output to be useful in seeing evolution over time. (Acts uses a metrics repository to store benchmark data points; no need for database.)

Technical implementation on eicweb

- eic_container: repository to build internal container with sandboxed reconstruction version (juggler, but eicrecon now also possible)
    - Any commit on any branch to reconstruction algorithms builds a dedicated container just to run benchmark
    - Could easily do the same with geometry
- common bench: utility repository to clone and compile geometry
    - Heavily reliant on eicweb local disk storage shared between jobs in the same pipeline (no GitHub Actions equivalent)
    - Mostly a bad idea, in hindsight

Why are users resistant to benchmarks?

Mostly a cultural or usability issue.

- Benefit is not clear to users: why be forced to write in a strict format when all that's expected is a set of plots for the next WG meeting?
  - Answer: We take over maintenance in the face of upgrades. E.g. tracking efficiency and resolution benchmark plots have been evolved from ATHENA through ePIC, constantly operational. We are in an international collaboration and we need to be able to re-produce and re-use our studies.
- Everything is hosted and run from eicweb instead of GitHub
  - Answer: We should come up with a structure that allows users to contribute and see results from benchmarks using GitHub.

Modularity

Tight integration between reconstruction framework and is not beneficial: benchmarks are different from unit tests of the reconstruction framework.

Benchmarks are more focused on determining the impact of the parameters that go into the reconstruction algorithms. Setting parameters effectively is a requirement on the reconstruction framework. (Reconstruction benchmarks on eicweb have many different reconstruction configurations that use just a subset of algorithms, run in debug mode, or use special parameter sets.)