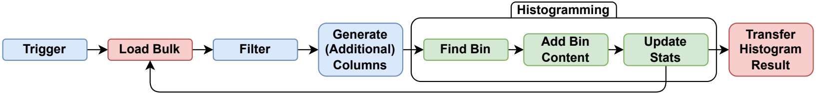
GPU Optimizations for HEP Analysis in ROOT

Kevin Nobel
Supervisors: Monica Dessole and Jolly Chen

Introduction



A typical HEP analysis:

- Load data
- Filter data
- Define additional columns
- Fill a histogram

Batch Histogramming Goal and Motivation

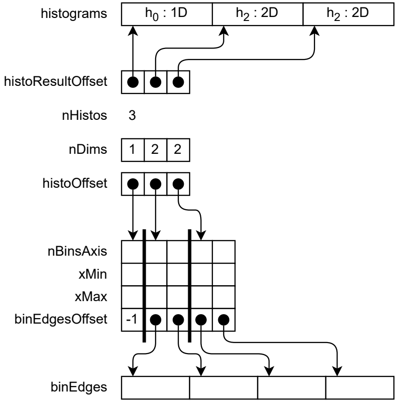
- Motivation: Future increase in data to be processed requires faster histogramming
- Goal: fill **multiple histograms** in parallel on GPUs
- Especially interesting if there is **overlap** in the input data
  - Better locality and a reduction in data transfers

Batch Histogramming Implementation

Features:

- Multiple histograms
- Mixed dimensions
- Different axis sizes
- Fixed and variable bins

GPU Datastructure →



Batch Histogramming
Results

- CPU: Ryzen 7 5700g, GPU: RTX 3060

- 1D + 2D + 2D histogram, 100 Million rows
- Speedup: 5.9× over single-threaded CPU impl.
- GPU `Fill` is much faster (417× speedup)
- Spend 98.6% of the (GPU) runtime on transferring data...

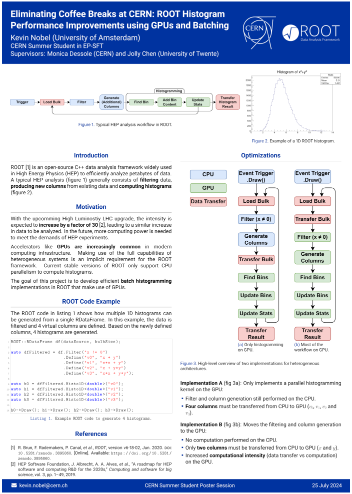
Offloading `.Define()` to GPUs

- Multiple histograms on the same columns are rare
- Generating new data based on the same columns is more common
- What if we compute new columns on the GPU?
  - Define may be computationally heavy
  - Potentially less data transferred

Poster Presentation

- Presented the idea at the Summer Student Poster Presentation[a]

[a] indico.cern.ch/event/1435014/



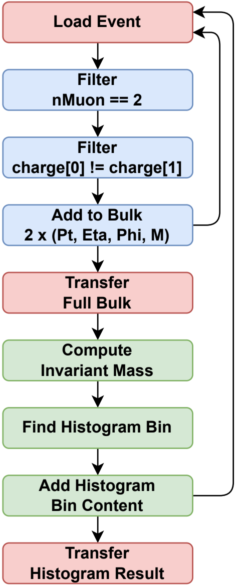
DiMuon

- Porting DiMuon analysis[1] from the ROOT tutorials
- Calculate invariant mass of all events with exactly 2 muons with opposite charge
- Not necessarily a good use-case
  - Just one histogram
  - Transfer 8 doubles to fill a single bin
  - Start with something simple

[1] root.cern/doc/master/df102__NanoAODDimuonAnalysis_8C.html

DiMuon Approach

- Discard irrelevant events on CPU
- Calculate invariant mass on GPU
- Fill histogram on GPU



DiMuon Results

- CPU: Ryzen 7 5700g, GPU: RTX 3060

- 24.067.843 events
- Speedup: 2.6× over 16 threads CPU impl.
- Data transfer: 57.5%

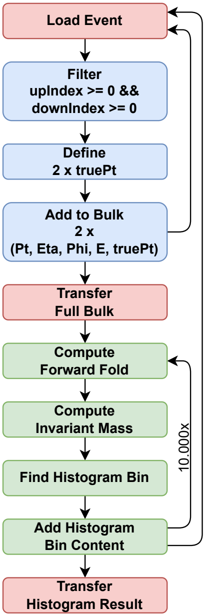
Folded W Mass

- 10.000 defines in a for loop
- All on just a few columns
- Varying scale + resolution for each define
  - 100 scales x 100 resolutions

Folded W Mass Approach

- Discard irrelevant events on CPU
- Calculate truePt values on CPU

- Apply forward folding on GPU
- Calculate invariant mass on GPU
- Fill histogram on GPU



Folded W Mass
Results

- CPU: Ryzen 7 5700g, GPU: RTX 3060

- Tested with 100.000 events (1 billion bins filled)
- Speedup: 95× over 16 threads CPU impl.
- Data transfer: 0.1%

Future Work

- Develop a generic solution to execute `.Define()` and `.Filter()` on GPUs
- Integrate GPU histogramming into ROOT

Conclusion

- Implemented a generic batch histogramming GPU kernel
- Explored how to potentially optimize `.Define()` on GPUs
- Presented my work during the Summer Student Poster Presentation

- Code is published on GitHub[2]

[2] github.com/tweska/cern-ssp/