Reconstruction directions

Giuseppe Cerati
Computing R&D Micro-Retreat
April 20, 2018

Introduction & Old/current model

Goal of this talk is to help the discussion, not meant to be a full review!

This is what could be today's typical model (of course there are exceptions!):

- Reconstruction code written by physicists (often non computing experts)
  - experts help with integration in framework and optimization of key components
- Code developed and run on CPUs (single thread, no vectorization)
- Workflows run at grid centers, one job per core
- Machine learning and multivariate tools used for tagging and selection of candidates/events
- Trigger: low level trigger on hardware/FPGAs, high level on CPU is a simplified version of offline reco

Major issues and challenges for the future

All are opportunities for better physics, we 'just' need to learn how to take advantage of them!

- Higher accelerator intensities: more data (both online and offline)
- New detectors: larger, more granular, new technologies
- End of Moore's law for frequency scaling, emerging of highly parallel architectures
  - parallelism both at SIMD/vectorization and thread level
- Heterogeneous systems
  - CPU/MIC/GPU/FPGA…, computing centers/supercomputers, …
- Big progress of deep learning techniques

Strategies we could pursue (I)

The points below are already being studied in R&D projects, the question is how a specific solution can become standard practice?

- Explore new architectures and parallelization (SIMD and thread) at sub-module level (algorithms)
  - vectorization is difficult but for more important than thread-level parallelization for CPUs
  - of course efforts for event level and module-level parallelization should continue

- Deep learning techniques
  - can machine learning be used for full event reconstruction?
  - or should we envision a heterogeneous reconstruction in terms of algorithms, e.g.:
    - ML for early event tagging, traditional algorithms for different reconstruction paths based on tag
    - ML for pattern reco, traditional fit
    - ...
  - how can we effectively input the physics to machine learning? so that training does not waste cycles on things we already know how to do, nor picks up unphysical solutions

Fermilab

Strategies we could pursue (II)

- Identify systems for processing of standard and non-standard workflows
  - on-demand reco on accelerators/co-processor: definitely for trigger, and also offline?
  - grid-like farms, supercomputers, commercial clouds, … ?

- Explore portable solutions and/or automatic code generation/optimization
  - same origin for different code on heterogeneous systems
    - identify data structures efficient for different architectures and models
    - what about auto-coding techniques? average physicist would write 'simple' code with algorithm flow, actual optimized code auto generated

- Development model may need to be revisited
  - centralized work gives higher quality code and consistent data products
  - distributed work gives access to more resources, closer to analyses
  - need a proper balance between the two modes!

… the question is how a specific solution can become standard practice?