GPU workflows in CMSSW

- currently (*e.g.* as of CMSSW_11_3_0_pre6) there are 4 representative workflows that can run (only) on GPUs
  - `#.502` pixel-only reconstruction with ”Patatrack” quadruplets
  - `#.506` pixel-only reconstruction with ”Patatrack” triplets
  - `#.512` ECAL-only reconstruction
  - `#.522` HCAL-only reconstruction
- and their corresponding CPU-only workflows
  - `#.501` pixel-only reconstruction with ”Patatrack” quadruplets
  - `#.505` pixel-only reconstruction with ”Patatrack” triplets
  - `#.511` ECAL-only reconstruction
  - `#.521` HCAL-only reconstruction
- running on CPU or GPU must be decided when the workflow is created

CC BY SA

redesigned workflows

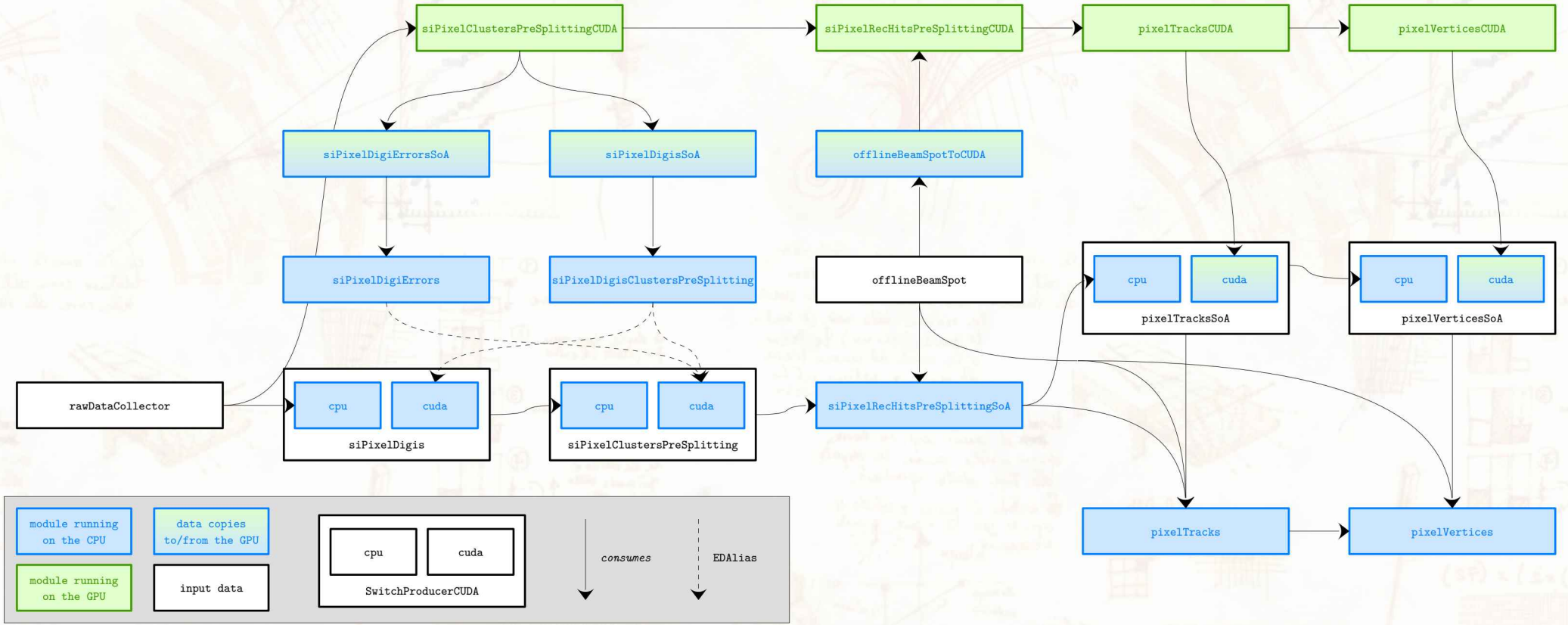
- as of #33428 / #33519 the GPU workflows no longer *require* a GPU
  - `#.502` pixel-only reconstruction with ”Patatrack” quadruplets
  - `#.506` pixel-only reconstruction with ”Patatrack” triplets
  - `#.512` ECAL-only reconstruction
  - `#.522` HCAL-only reconstruction

- these workflows now detect if a GPU is available at runtime, and adapt accordingly
  - based on the `SwitchProducerCUDA` mechanism
  - choose at job startup whether to follow the `cpu` branch or the `cuda` branch

- the corresponding CPU workflows use the same configuration
  - but implement only the `cpu` branch of the switches

CC BY SA

pixel ntuplets workflows (#.502/#.506)



next steps (for future PRs)

- add workflows that consume *both* CPU and GPU products
  - useful for DQM and validation of GPU vs CPU results
  - should be reasonably easy, by explicitly consuming both branches of the `SwitchProducerCUDA`s
- implement a switch that forces the reconstruction to run on CPU or on GPU
  - *i.e.* a workflow does not use a GPU even if it is available, or fails if no GPUs are available
- do we want to keep the current CPU-only workflows (*e.g.* `#.501`) ?
  - or replace them with running the GPU-optional workflows (*e.g.* `#.502`) without a GPU ?
- add a single workflow that runs all producers
  - pixel tracks and vertices, ECAL local reconstruction, HCAL local reconstruction

  which should be useful for various developments
  - speed up validation and benchmarks
  - continue the development of reconstruction modules, *e.g.* pixel-based particle flow, electron seeding