

Hydra

Status report and prospects for the 2.0 release.

A. Augusto Alves Jr

Presented at DIANA meeting, CERN, September 11, 2017



- **Highlights**
- **Status**
- **New features**
- **Summary**

Hydra is a header only templated C++ library designed to perform common tasks found in HEP data analyses using massively parallel platforms.

- It is implemented on top of the C++11 Standard Library and a variadic version of the Thrust library.
- Hydra is designed to run on Linux systems and to use OpenMP, CUDA and TBB enabled devices.
- It is focused on portability, usability, performance and precision.

The main design features are:

- The library is structured using static polymorphism.
- The library provides a set of optimized containers that can store multidimensional data-sets efficiently using SOA layout.
- Algorithm and data separation is enforced. Data handled using iterators and all classes manages their resources using RAI.
- Library is type and thread-safe. Objects handled by the interface are deeply inspected and usually code that would clash at run-time due type mismatches or stack overflow does not compile.

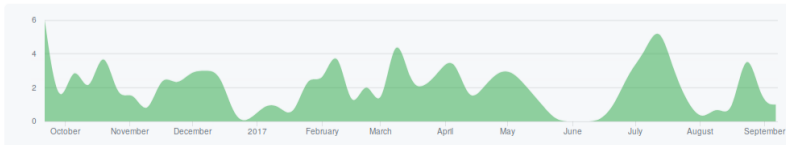
The source files written using Hydra and standard C++ compile for GPU and CPU just exchanging the extension from `.cu` to `.cpp` and one or two compiler flags. There is no need to re-factory code.

Status of the code currently available on the Hydra official repository:

<https://github.com/MultithreadCorner/Hydra>

- Code corresponds to the a snapshot of the project status from about 10 months ago. Last commit on March/2017
- This release is functional and stable. Total of 51 commits.
- Development effort has been active the @AAAlvesJr's development fork: <https://github.com/AAAlvesJr/Hydra>. Total of 356 commits by now.
- Many new algorithms and major project code restructure.

Contributions to master, excluding merge commits



- Generic multidimensional containers: `hydra::multivector` and `hydra::multiarray`
- Numerical integration: adaptive and non-adaptive Gauss-Kronrod quadrature, non-adaptive Genz-Malik quadrature.
- Phase-space generation: new interface supporting integration and evaluation of models “in place”, re-weighting and un-weighting of samples using user-defined functors, new and faster container for events. Generator now checks for kinematic consistence between mother and daughter particles.
- New interface for `ROOT::Minuit2`. Fits can now be performed using lambdas on host and device side.
- New class to calculate S-Plots.

- All supported back-ends can run concurrently in the same program using the suitable policies: `hydra::omp::sys` , `hydra::cuda::sys`, `hydra::tbb::sys` , `hydra::cpp::sys` , `hydra::host::sys` and `hydra::device::sys`.
- Project does not depend on an external Thrust installation anymore. Hydra 2.0 will have its own version of Thrust and also of Eigen.
- New set of examples, distributed in directories by functionality.
- Plan to provide performance measurement suite using Nonius.

- The merging of the new code in the stable repository is imminent (about two weeks).
- The new stack will be put initially in pre-release status, to get synchronized with Hydra.Python.
- Hydra and Hydra.Python new releases will happen together.