C++ as a service — rapid software development and dynamic interoperability with Python and beyond

Interactive C++: cling and clang-repl

Vassil Vassilev

03.06.2021
Add support for numerical differentiation fallback in Clad

Modernize the LLVM “Building A JIT” Tutorial Series

Utilize second order derivatives from Clad in ROOT

Add support for differentiating functor objects in clad
The initial version of clang-repl landed! [D96033]
Status. Cling

- Reduced patches from our clang fork
  - D77598 (Integral template argument suffix printing) [Pratyush] — Landed in llvm but needs tuning to replace the patch in ROOT.

- Investigate crashes in Cling with the Thrust library — resolved by avoiding to rely on Thrust so far
Status. InterOp

- Second revision of the interoperability spec (requires some cleanup).
- Thanks to Wim we have defined a path forward with a deliverable basic template instantiation support when bridging python and clang-repl. The target is early fall.
- This work will also polish the requirements doc further and show code examples.
Status. Clad

- Resolve issues with Clad argument passing — clarified.
- Released Clad v0.8
- [Baidyanath] Working on proper tapenade-style array support in clad
- [Garima] Advances the user-extensible error estimation framework. Will work on preparing a generic clad tutorial.
Plans

- Start rebasing cling to LLVM13.
  - That would require a clad-like compatibility layer to maintain the llvm9 support
- Work on the Partial Translation Unit design for clang-repl
- Accelerate upstreaming clang patches
- Automatically differentiate the CUDA kernels (including computation scheduler)
- Enable Clad in xeus-cling
CaaS Open Projects

- **Patches against clang.git**
  - Implement FileManager uncaching
  - Adapt the user of invalidateCache to its new signature
  - Mark the file entry invalid, until reread
  - Propagate cache flags from LookupFile() to FileManager::getFile()
  - Pass the OpenFile flag also to DirectoryLookup
  - Do not load the source file just to get an irrelevant SourceLoc (ROOT-7111)
  - Allow interfaces to operate on in-memory buffers with no source location info [Pratyush Das]

- Open projects are tracked in out open projects page.
Next Meetings

- Monthly Meeting — 1st July, 1700 CET/0800 PDT
- Ioana and Garima will talk about their projects in an IRIS-HEP topical meeting on 21st of June. Details [here](#).
- Tentative talk schedule:
  - OrcV2, Lang Hames, Apple, July
  - Language Interop Progress, Vassil, Princeton, Aug
  - Cppyy — how to bridge dynamically python and C++, Wim Lavrijsen, LBL, Sep

If you want to share your knowledge/experience with interactive C++ we can include presentations at an upcoming next meeting.
Thank you!