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

Interactive C++: cling and clang-repl

Vassil Vassilev

01.07.2021

Status. Clang-Repl

- The initial version of the partial translation unit approach to error recovery was accepted! [D104918]

Status. Cling

- Reduced patches from our clang fork
  - D103040 (Print default template argument if manually specified in typedef declaration.) [Pratyush]
- Started rebasing cling on top of llvm13
- Investigated how to enable clad in xeus-cling

Status. InterOp

- Started working on a python reference example provided by Wim

Status. Clad

- Improvements in documentation, bug fixes, forward mode differentiation of functor objects
- [Baidyanath] Working on proper tapenade-style array support in clad and implementing the changes in the clad::hessian matrix.
- [Garima] Advances the user-extensible error estimation framework. Will work on preparing a generic clad tutorial. Working on adding a numerical differentiation fallback
- [Parth] Prepares a tutorial on clad troubleshooting for developers. Improved functor support for reverse mode.

Plans

- Speed up the interoperability work
- Submit abstracts on automatic differentiation and interactive C++ for CppCon.
- Prepare a paper about the work we've completed.
- Enable error recovery for advanced C++ code (eg template instantiation)
- Accelerate upstreaming clang patches
- Automatically differentiate the CUDA kernels (including computation scheduler)
- Enable Clad in xeus-cling by default

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 — 5th August, 1700 CET/0800 PDT
- Tentative talk schedule:
  - 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!