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

Interactive C++: cling and clang-repl

Vassil Vassilev

05.08.2021

Status. Clang-Repl

- Clang-repl will be shipped with llvm13
- Submit abstracts on automatic differentiation and interactive C++ for CppCon.

Status. Cling

- Reduced patches from our clang fork
    - D103040 (Print default template argument if manually specified in typedef declaration.) [Pratyush]
- Continuing to rebase cling on top of llvm13 — we need to split the monorepo commits in our forks llvm.git and clang.git
- Worked towards xeus-clad combining clad and xeus-cling conda packages

Status. InterOp

- Work on template instantiations on demand is complete
- Thanks to Wim, we have a minimal version of python script to trigger the instantiation. We need a ctype-based wrapper library for the template instantiation code to keep things minimal

Status. Clad

- Improvements in documentation, 3 tutorials ready, bug fixes
- [Baidyanath] Array support is done!
- [Garima] Several iterations of the error estimation framework (including the rebase to use array support). Progress on adding a numerical differentiation fallback
- [Parth] Functor support is complete. Implemented differentiation of while and switch statements in forward mode. Preparing a tutorial on clad troubleshooting for developers. Working on documentation, bug fixing.

Plans

- Speed up the interoperability work
- 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)

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 our open projects page.

Next Meetings

- Monthly Meeting — 2nd September, 1700 CET/0800 PDT
- Tentative talk schedule:
  - Cppyy — how to bridge dynamically python and C++, Wim Lavrijsen, LBL, Sep
  - Language Interop Progress, Vassil, Princeton, Oct
  - LLDB, Raphael Isemann, Apple, Nov

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

Thank you!