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

Interactive C++: cling and clang-repl

Vassil Vassilev

25.03.2021

Status

- LLVM9 Upgrade
  - ROOT works well (still a few unrelated blockers)
  - CMSSW Modules IB is almost green [David]
- Landed patches
  - https://reviews.llvm.org/D91524 (safe lookup on deserialization)
- Progress on clang patches:
  - Working on https://reviews.llvm.org/D41416 (lazy pcm template deserialization)

Status

- Cling Release
  - Green light from ROOT
  - Green light from cling's test suite
  - Green light from the CUDA backend (Thanks Simeon)
  - Still need to hear from xeus-cling
  - A remaining issue in cpt.py
- Cling now builds its plugins by default (eg. clad is available).

Status

- Proposed a very limited version of clang-repl following the design of cling: [D96033]. There are more comments in the area of ABI and CodeGen.
- Preparing for the Google Summer of Code 2021
  - Several very promising candidates
- Early stage technical specification of the language interoperability layer — here.
  - Started looking into boost::describe which may be the way to go. Meanwhile, additional comments are welcome.

Status

- Finished a CUDA - Clad integration script which takes a user defined function and calls Clad to compute the derivative with respect to some chosen variable and passes the resulted function on the GPU [Ioana]
- Included the changes required for this script in Clad, i.e. cloning the function attributes with respect to the CUDA context (__device__ __host__) [Ioana]
- Our blog post PR is submitted (Many thanks everybody who contributed and in particular to Wim Lavrijsen and Alexandru Militaru)

Plans

- Accelerate upstreaming clang patches
- Release Cling and Clad
- Second revision of the interoperability spec
- Understand CUDA failed to copy a symbol to device only within Cling (works in Clang9) [Ioana]
- Resolve issues with Clad argument passing (“-fdump-derived-fn” combined with “fdump-source-fn-ast”) [Ioana]

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 — 6th May, 1700 CET/0800 PDT
- Tentative talk schedule:
  - cppyy, Wim Lavrijsen, LBL, May
  - Clad/CUDA, Ioana Ifrim, Princeton, May
  - OrcV2, Lang Hames, Apple, June

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

Thank you!