

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

Interactive C++: cling and clang-repl

Vassil Vassilev

---

10.03.2022

# Status. Cling

---

- ❖ Continuing to rebase cling on top of llvm13 (40 tests out of 185 fail)

# Status. Clang-Repl

---

- ❖ Built a quick prototype that can connect python to clang-repl in clang-14-rc to be potentially used in the awkward array project connecting Python and C++
- ❖ We will collect first round of feedback and we can prepare a demo

The goal is to provide an easy installation and use of the Cling-inspired incremental compilation we landed in Clang

# Status. InterOp

---

- ❖ Currently working on improving the performance of CreateScopeProxy in cppyy. Need to coordinate the work with the D use-case
- ❖ Investigating how to use llvmlite to generate LLVM IR in Numba which calls into cppyy

The goal is to rework the python-to-C++ automatic binding generator cppyy to use LLVM interfaces which can help improving speed and accuracy

# Status. Clad

---

- ❖ Made Clad's custom derivatives utilize the pushforward mechanism
- ❖ Finished support for passed by reference arguments and return types
- ❖ Implemented single pass pushforward support (computing the value and its gradient)
- ❖ Completed support of derivatives of user defined types wrt scalar builtin types (eg `std::complex`)
- ❖ Finished the paper for ACAT on GPU Accelerated AD
- ❖ Added [binder](#) tutorial for clad
- ❖ Working on various prototypes and have a concrete approach for getting Clad derivatives from RooFit functions

# People

---



Purva Chaudhari  
Error Recovery in  
Clang-Repl  
(Feb 2022-May 2022)

# 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 — 7th Apr, 1700 CET / 0800 PDT
  - ❖ Matheus Izvekov — Presentation on type re-sugaring in clang

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



Thank you!