

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

**Interactive C++: cling and clang-repl**

Vassil Vassilev

---

07.04.2022

# Status. Cling

---

- ❖ Continuing to rebase cling on top of llvm13 (20 tests out of 185 fail)
- ❖ Opened a pull request against ROOT (between 60 and 160 test fail)

The goal is to move the infrastructure to llvm13 and start reusing the deliverables from the project which are available in clangInterpreter.

# Status. Clang-Repl

---

- ❖ Making progress on error recovery verification
- ❖ Implemented a special recovery mode and ran various clang test cases in recovery mode where we include and undo the test.
- ❖ More than 200 tests just work and we are investigating around 10 failures

The goal is to provide a more stable error recovery approach than the currently available one in clang

# Status. InterOp

---

- ❖ The conclusion from the discussion with the D use-case is that we should provide a lookup-based interfaces and iteration-based ones (Breadth-first)
- ❖ Currently working on simplifying the `meta_getattro` function
- ❖ Investigating how to use `llvmlite` to generate LLVM IR in Numba which calls into `cppy`

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

# Status. Clad

---

- ❖ Added support for differentiating constructors
- ❖ Still working on kernel differentiation
- ❖ Worked on demonstrating floating point error estimation using AD and Clad
- ❖ Improved the documentation of Clad

# 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 — 5th May, 1700 CET / 0800 PDT

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

Thank you!