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

Interactive C++: cling and clang-repl

Vassil Vassilev

### Status. Cling

- Continuing to rebase cling on top of llvm13
- \* The ppc64 support for Cling is back (we lost it in llvm8).

## Status. InterOp

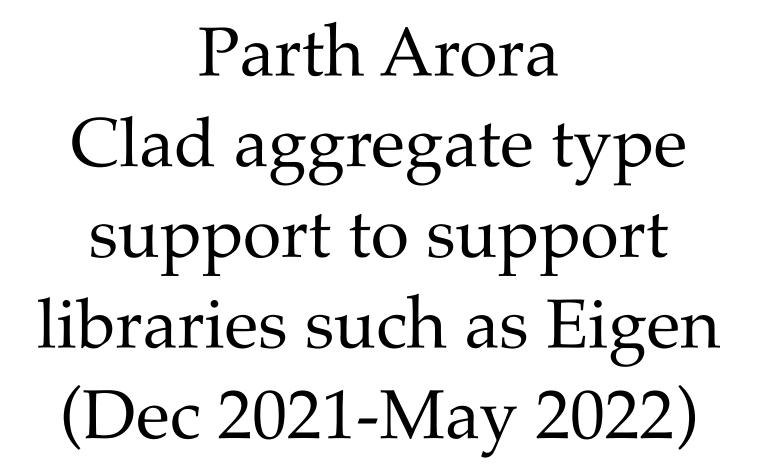
- \* The <u>document</u> is ready. We are looking forward to your feedback.
- Addressed several comments and still some minor improvements but mostly happy with the current state.

### Status. Clad

- \* A <u>talk</u> by Ioana on "Automatic Differentiation for C++ and Cuda using Clad" at ACAT
- Poster on NeurIPS about Clad

### People











Baidyanath Kundu cppyy, libInterOp (Jan 2022-Dec 2022)

### Plans

- 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 — 13th January, 1700 CET/0800 PDT

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

