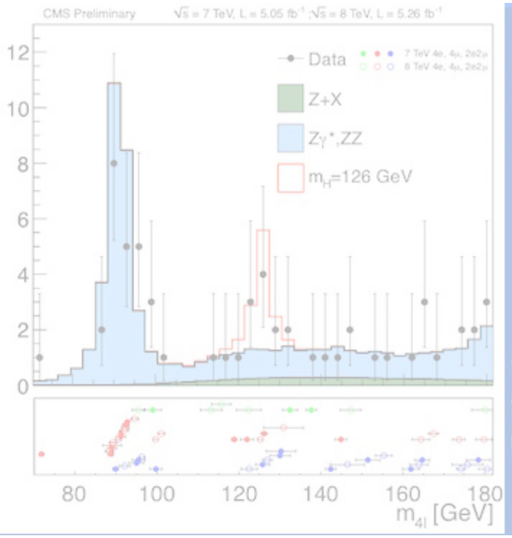

Compiler Research

Status And Plans

Vassil Vassilev

04.04.2024

Clad — Enabling Differentiable Programming in Science

Source Transformation AD With Clad

- Initial support of memory operations in reverse mode
- Added support for clang-18 (discovered regression in clang in #87151)
- Re-enabled tests on 32-bit
- Simplified pullbacks in reverse mode (produces 10% less code)
- Delayed the differentiation process to the end of the translation unit
- Reduced excessive stores in the error estimation mode.
- Reduced clad::array_ref and clad::array usage in the generated code
- Next milestone v1.5 is delayed due to the substantial changes introduced to Clad and the possibly breaking change in the gradient forward declarations

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

Hands on details can be seen in our showcase presentation.

Status. Cling

- No updates

Status. Clang-Repl

- 7 merged contributions last month: link
- 8 contributions updated this month: link
- Value Handling (RFC)
  - D146809 — [clang-repl] Implement Value pretty printing for containers

The goal is to provide better stability and robustness which can later cling can reuse.

Status. CppInterOp

- Improved CMake config that makes find_package more robust
- Added tests in the conda package via clad-feedstock
- Added arm packaging support
- Working on adding advanced template instantiation support

Status. Xeus-Cpp

- CppInterOp v1.2.0 was integrated in xeus-cpp PR46
- Working on merging more infrastructure xeus-clang-repl into xeus-cpp
- Released v0.4.0

Status. Xeus-Clang-Repl

- No updates

Open Projects

- Open projects are tracked in our open projects page.

Next Meetings

- Monthly Meeting — 2nd 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!

Lingo

- **CppInterOp** is a product of OAC-1931408 and exposes API from Clang and LLVM in a mostly backward compatibe way. The API support downstream tools that utilize interactive C++ by using the compiler as a service. That is, embed Clang and LLVM as a libraries in their codebases. The API are designed to be minimalistic and aid non-trivial tasks such as language interoperability on the fly. In such scenarios CppInterOp can be used to provide the necessary introspection information to the other side helping the language cross talk. The package makes it easy to deploy as it ships Clang as a service without any dependencies.
- **Xeus-Clang-Repl** is a product of OAC-1931408 that is a Jupyter plugin supporting C++ development based on ClangRepl.
- **Xeus-Cpp** is a product of OAC-1931408 in collaboration with the QuantStack company. It is a Jupyter kernel for C++ based on the native implementation of the Jupyter protocol xeus. It is supports the Wasm version of Jupyter – JupyterLite. Generalization of Xeus-Clang-Repl.

Lingo

- **Cling** The first C++11-compliant interpreter used in the field of High-Energy Physics for data analysis and interoperability.
- **ClangRepl** is a generalization of Cling in LLVM/Clang upstream and is a product of OAC- 1931408. It be more reliable, easier to deploy. It follows the best practices adopted by the LLVM developers community. It supports CUDA, OpenMP and Wasm.
- **Cppyy** is an undervalued, cutting-edge Python/C++ language interoperability tool originated by Wim Lavrijsen, LBL. It is the de-facto standard for efficient Python/C++ interoperability in the field of particle physics. As part of OAC-1931408 our group collaborated with LBL improve packaging and reduce the dependencies allowing cppyy to move closer to LLVM orbit.