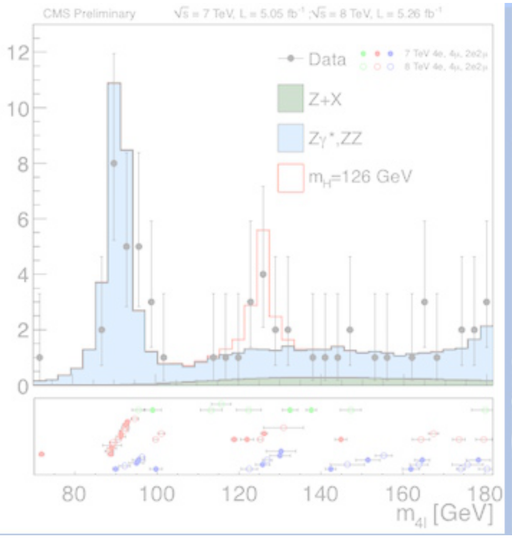

Compiler Research

Status And Plans

Vassil Vassilev

11.01.2024

Clad — Enabling Differentiable Programming in Science

People

Vaibhav Thakkar

GSoC22, Electrical Engineering and Computer Science, Indian Institute of Technology, Kanpur, India

Clad.
Research Intern @ CERN

Aaron Jomy

GSoC23, B. Tech in Computer Science, Manipal Institute of Technology, Manipal, India

Cppyy & CppInterOp
Research Intern @ CERN

Source Transformation AD With Clad

- Establishing collaborations with stakeholders
- Discussing milestones and deliverables
  - Defined 5 milestones, v1.3, v1.4, v1.5, v1.6 and v1.7:

Source Transformation AD With Clad

- Processed a few long standing issues #687, #275, #86, #49, #650
- Improved the CI especially when debugging remote bot failures
- Fixed the strong symbols in Differentiator.h
- Improved the spelling of cmake options
- Added support for non-type template parameters
- Added basic infrastructure to support Kokkos
- Improved documentation on how to build clad for developers
- Started preparing for the next GSoC round.

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

- 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. Xeus-Clang-Repl

- No updates

Status. Xeus-Cpp

- No updates

Status. CppInterOp

- Enabled Windows builds, still several issues to be resolved
- Increased the testing coverage

Open Projects

- Open projects are tracked in our open projects page.

Next Meetings

- Monthly Meeting — 7th March, 1700 CET/0800 PDT
  - Tentative — Interactive Differential Debugging

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.