



# Compiler Research

## Status And Plans

Vassil Vassilev

06.06.2024

# People

---



Maksym  
Andriichuk

*GSoC24, University of Wuerzburg, DE*

Optimizing automatic  
differentiation using activity  
analysis

[Info](#)

# Clad — Enabling Differentiable Programming in Science

---



# Source Transformation AD With Clad

---

- ❖ v1.6 is almost ready to go
  - ❖ Added support for `std::string`
  - ❖ Added support for lambda expressions (no captures yet)
  - ❖ Added support for computing the hessian diagonal only
  - ❖ Various refactorings to prepare the codebase for larger modifications
  - ❖ Fixes wrt the ATLAS Higgs combination benchmarks and CMS combine Higgs L1 analysis
- ❖ Scientific use-cases
  - ❖ RooFit's Clad-based ATLAS Higgs combination benchmark works and scales well.
  - ❖ CMS combine Higgs analysis with CMS open data is almost complete.
  - ❖ More in Vaibhav's presentation and ICHEP.
- ❖ Next milestone v1.7 is planned in the end of the month

# 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

---

- ❖ Being upgraded to llvm18.

# Status. Clang-Repl

---

- ❖ 3 merged contributions last two months: [link](#)
- ❖ 2 contributions updated last two months: [link](#)
- ❖ [PR86402](#) — [clang-repl] Support wasm execution has landed!
- ❖ Making slow progress on:
  - ❖ [PR84769](#) — [clang-repl] Implement Value pretty printing for containers. Value Handling ([RFC](#))

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

# Status. CppInterOp

---

- ❖ Working on enabling automatic library loading. PR 308
  - ❖ This is the last missing element to deprecate completely xeus-cling in favor of xeus-cpp



# Status. *Xeus-Cpp*

---

- ❖ Working on adding LLM support
- ❖ Working on merging more infrastructure `xeus-clang-repl` into `xeus-cpp`
- ❖ Released v0.5.0
- ❖ Releasing a major release to deprecate `xeus-cling` requires 1 feature to be implemented wrt automatically loading of symbols.

# Status. *Xeus-Clang-Repl*

---

❖ No updates

# Open Projects

---

- ❖ Open projects are tracked in our [open projects page](#).

# Next Meetings

---

- ❖ Monthly Meeting — 1st Aug, 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 compatible 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 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.