Compiler-Research: State of the Union



VASSIL VASSILEV

Princeton University · CERN

CompilerResearchCon 2026 · 3 November 2026

The year since CRCon25

1,178

merged pull requests

in our repositories · 797 the year before

354

upstream pull requests

LLVM, ROOT, CMS, … · 204 the year before

12

releases

Clad, CppInterOp, xeus-cpp, Cling, cppjit, RAMTools

60

contributors

40 of them new · 39 the year before

19

CRCon26 talks scheduled

8 at CRCon25

32

blog posts

plus 16 talks and training sessions

What we build

Clad

Automatic differentiation for C, C++ and CUDA as a Clang plugin

Cling & clang-repl

Interactive C++: the ROOT interpreter, upstream in LLVM

CppInterOp

C++ reflection and interop layer used by cppjit, xeus-cpp and jank; being adopted in ROOT

cppjit: our flagship

Automatic Python-C++ bindings; the successor of cppyy

xeus-cpp

Jupyter kernel for C++, including in the browser via WebAssembly

RAMTools & CARTopiaX

Genomics on ROOT RNTuple; CAR-T cancer therapy simulation on BioDynaMo

What drove this year's output



Clad is being integrated seriously in HEP

Clad: 493 merged pull requests (347 the year before) from 30 contributors besides the lead (20) · CMS Combine: 40 upstream pull requests (22)

AI writes a growing share of the code

The lead's 515 merged pull requests (118 the year before) are largely AI-generated and reviewed by him · 47% of all merged pull requests had no second reviewer

Experienced reviewers are now the bottleneck

Code is getting cheaper; we need more people who can review and merge, in every project

Technical highlights

Clad 2.5

Readable derivatives: real source locations, so a debugger steps through generated code

GPUs in Clad

Consolidating Clad's GPU infrastructure (Vedant Goyal); CUDA support in CppInterOp 1.9

CppInterOp 1.8 and 1.9

Extended JIT and incremental compilation; new API dispatch; hardened interpreter

Clang, by our fellows

ClangIR AMDGPU builtins, C2y language features, new warnings, clang-tidy and clangd fixes, front-end performance

clang-repl and ORC JIT

Automatic shared-library resolution of unresolved symbols in LLVM's ORC JIT (Sahil Patidar); earlier declarations survive a failed input (Vipul Cariappa)

12 releases since CRCon25

cppjit 0.1.0a1 · RAMTools 0.1.0 with a citable DOI · Clad 2.2-2.5 · CppInterOp 1.8 and 1.9 · xeus-cpp 0.8.1-0.10.0 · Cling 1.3

Flagship: cppjit

From a 2019 plan to a 2026 release: cppyy rebuilt on CppInterOp and clang-repl

2019: the plan

Long-term vision set with the NSF award C++ as a service (OAC-1931408), which created clang-repl

2023: CppInterOp 1.0

The C++ interop layer to migrate onto

2023-2026: 563 pull requests

in our cppyy, CPyCppyy and cppyy-backend forks, moving cppyy onto CppInterOp and clang-repl

Sep 2026: cppjit 0.1.0a1

The successor of cppyy, already in production at Chicago Trading Company

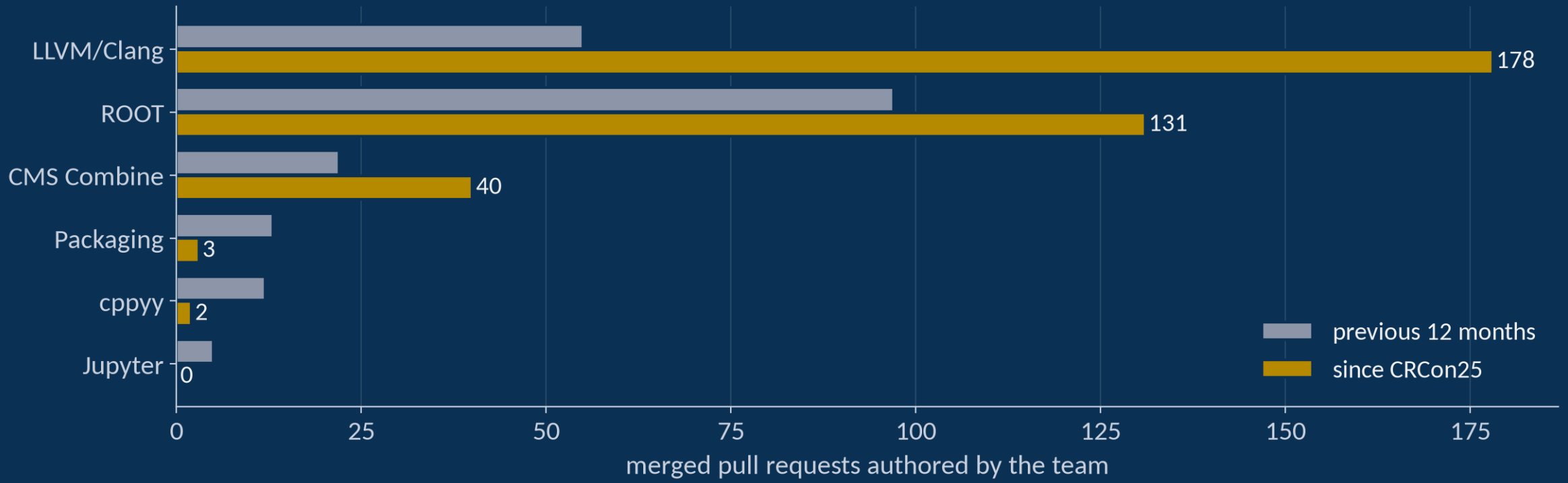
Kudos to the migration team

Aaron Jomy, Vipul Cariappa, Baidyanath Kundu, Matthew Barton, Jonas Rembser, Emery Conrad, Alexander Penev, Kerem Şahin, and 3 community contributors

and Wim Lavrijsen, whose cppyy it carries forward: cppyy's repositories now point to ours



Upstream impact



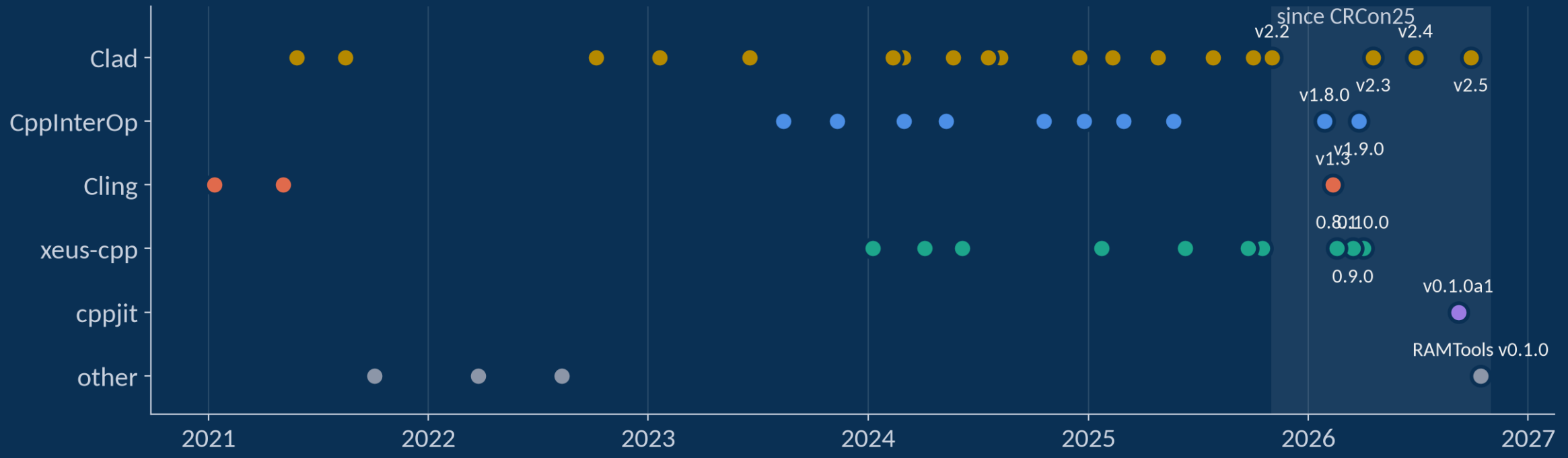
354 team-authored pull requests merged upstream since CRCon25, up from 204

Our 7 C++ Alliance fellows merged 93 LLVM pull requests since the fellowship began on 15 May, reviewed 63 others' and filed 10 issues · ROOT counts only interpreter, interop and AD work

Last year's commitments

GPU reverse mode at scale *RSBench and LULESH as acceptance tests*	Clad benchmarks: LULESH and XSBench merged, RSBench in review; GPU offload for the tape; CUDA reverse-mode fixes (Vedant Goyal); CppInterOp 1.9 adds CUDA support	**About 60-70% done** LULESH and XSBench are in, RSBench nearly done; the scattered GPU code has been consolidated (Vedant Goyal)
Packaging and cross-platform reliability *macOS, Windows and dependency churn*	New shared CI and prebuilt LLVM toolchains for Linux, macOS, Windows and WebAssembly (139 merged pull requests in ci-workflows)	**Done** the shared CI and toolchains in ci-workflows
Shared JIT and interop hardening *Symbol resolution, thread safety, long-running sessions*	CppInterOp 1.8 and 1.9 harden incremental compilation and dispatch; clang-repl symbol resolution; cppjit 0.1.0a1	**In production** deployed at CERN and Chicago Trading Company; used by jank; fixes upstreamed; cooperation with Apple engineers

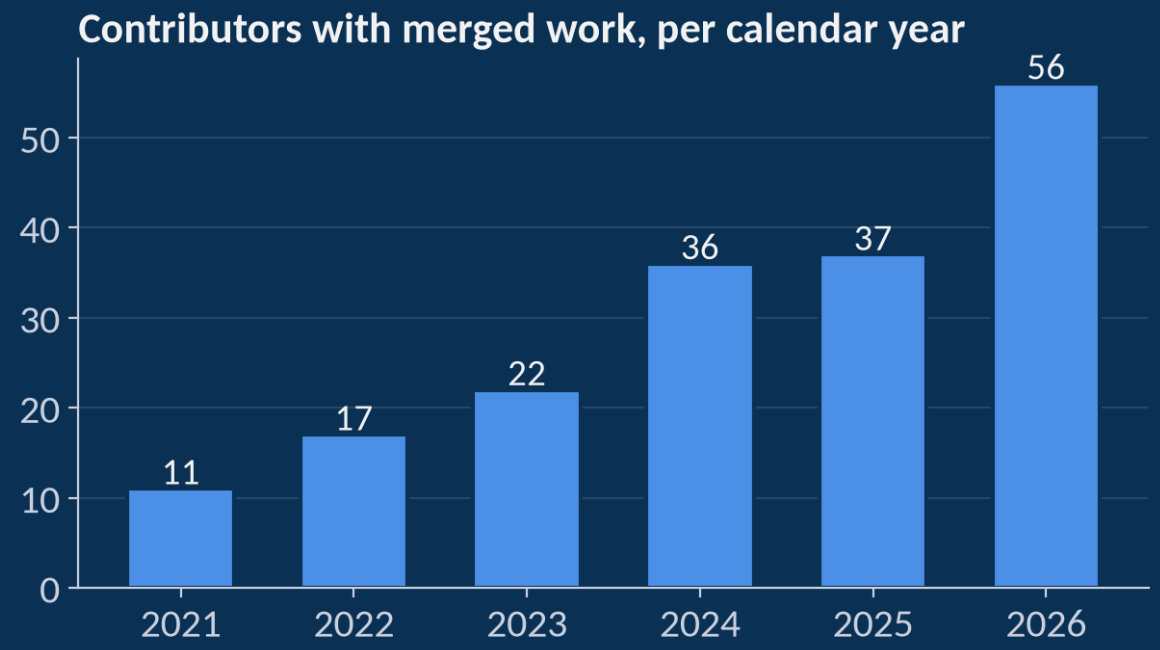
Released Software



12 releases since CRCon25

cppjit 0.1.0a1, the cppyy successor built on CppInterOp and clang-repl · RAMTools 0.1.0 with a citable Zenodo DOI · Clad 2.2-2.5 · CppInterOp 1.8 and 1.9 · xeus-cpp 0.8.1-0.10.0 · Cling 1.3

Organization health





5 h
median first review of contributors' PRs

23 h
median time to merge contributors' PRs

47%
of merged PRs had no review by a second person

40
new contributors

Risks and gaps

Funding cliff

Our NSF funding runs out in February 2027 (award OAC-2311471 formally ends in September); programs run on GSoC, C++ Alliance and volunteer time

Review capacity

66% of merges go through one reviewer (95% in Clad). Reviews are still fast (first review in 5 h, merge in 23 h), but AI raises output faster than review capacity

Publications

NSF records one journal article since 2024 (Computer Physics Communications, 2025); the year's results above are mostly in blog posts and talks

Retention

49% of last year's contributors are active again this year

→ To keep this going beyond any one grant, we propose to formalize what already works.

Impact across domains

High-energy physics

RooFit + Clad: up to 10x faster likelihood fits · automatic differentiation in CMS Combine · Cling is ROOT's interpreter; ROOT is adopting CppInterOp

Industry

Chicago Trading Company: Clang-REPL, CppInterOp and cppjit in production, fixes go back upstream · QuantStack: xeus-cpp and C++ in the browser

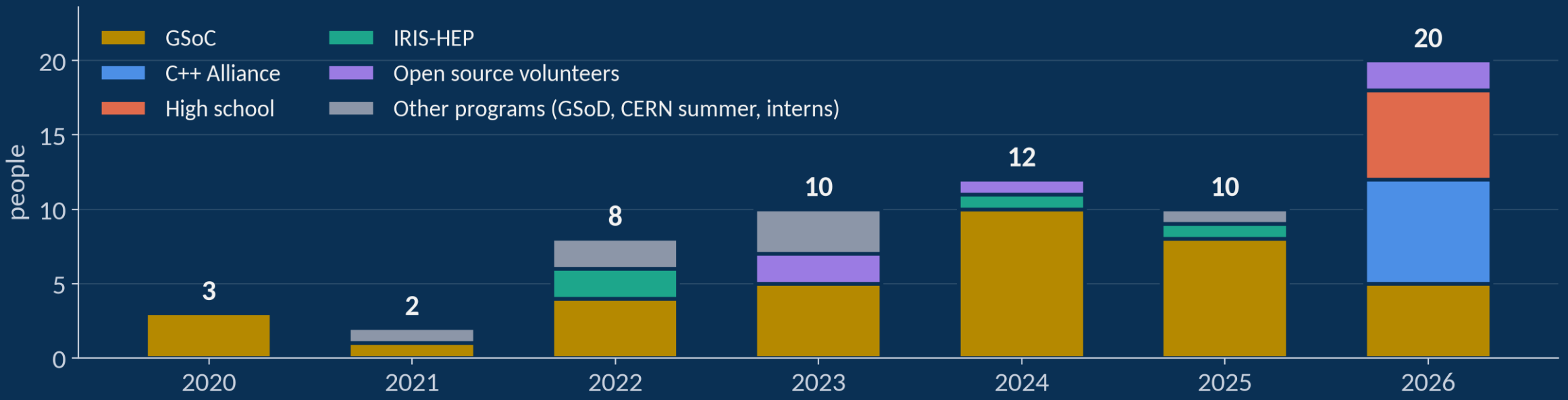
Life sciences

CARTopiaX: CAR-T cancer therapy simulation on BioDynaMo, reproduces the key trends of a wet-lab experiment · RAMTools: genomics on ROOT RNTuple, 0.1.0 with a citable DOI

HPC and training

Hands-on Accelerated Python tutorials at ISC 2026 and SC26 (16 November), built on NVIDIA's Accelerated Computing Hub · NVIDIA at CERN meeting, July 2026

People: a growing pipeline



11 projects completed

with merged work · 8 in progress · 5 not completed

6 high-school interns

10th graders from Plovdiv; 2 with merged code so far

Outside our programs

31 people got 92 pull requests merged

CompilerResearchCon

CRCon25

2 days

8 talks

CRCon26

3 days

19 talks scheduled

industry, academia, students

CRCon27

November 2027

Aiming for hybrid, with an in-person component

In-person venue to be decided; CERN under consideration



Toward a Compiler Research Foundation

Formalize what already works

Fellowships & mentoring
GSoC, C++ Alliance, high-school interns

Events & training
CRCon, ISC and SC tutorials

Shared infrastructure
CI, packaging, releases

Projects govern themselves
the foundation supports, it does not command

A long-term home, independent of any one person, employer or company

Steward, not owner: copyright stays with the authors, everything under open licences

Public benefit, no profit distribution · international and transparent by design

How to help

Review and merge

Experienced reviewers with merge rights, in every project

Try cppjit 0.1.0a1

Use it on your bindings and tell us what breaks

CRCon27

Hosts and partners for the in-person part of a hybrid event

Write it up

Co-authors to turn this year's results into papers

Bring people in

GSoC, C++ Alliance fellowships, high-school internships: mentors and projects

Tell us where it hurts

Packaging and platform problems, with a reproducer



Thank you

To all 60 contributors since CRCon25, and every mentor, supervisor and partner

@a-elrawy @Aadeshveer @aaronj0 @aditimjoshi @AdityaPandeyCN @aminatpwk @aniruddhaadak80 @AnonMiraj @anutosh491 @arima-04 @atharrva01 @ayokunle321 @ayush4874 @bassiounix @bellenot @codegallivant @conrade-ctc @devajithvs @Elvand-Lie @Errant404 @EuclidDivisionLemma @fatfat123-arch @Fermats-Last-Theorem @flash1729 @fogsong233 @GeorgiLH @guitargeek @hahnjo @hardikxk @hredayshah2507 @HrisShterev @jjerphan @jmcarcell @JohanMabille @keremsahn @kr-2003 @leetcodez @Lucky-Lodhi2004 @manpreetsinghraina191-rgb @mcbarton @MdSaifAliMolla @muskan-creates352 @mvanhorn @mvassilev @neo-0007 @ovdiiuv @PetroZarytskyi @r-timmaraju @RedBlueBird @rybkine @SahilPatidar @salva24 @ShresthSamyak @SylvainCorlay @tryh4rd-26 @Vedant2005goyal @venusflytrapfairy @vgvassilev @Vipul-Cariappa @voyager-jhk

With support from NSF · C++ Alliance · Google · Princeton · CERN · Chicago Trading Company · QuantStack · Cancer Research UK · University of Plovdiv · Math High School Plovdiv · IRIS-HEP · NVIDIA

Get involved: GSoC and fellowships · good first issues · CRCon27 · compiler-research.org

Backup: how we counted

Window: 1 Nov 2025 - 31 Oct 2026, compared with the 12 months before

Our repositories: Clad, CppInterOp, xeus-cpp, cppjit, Cling, CARTopiaX, RAMTools, ci-workflows, tutorials. All merged pull requests count; bots excluded.

Upstream: LLVM/Clang, ROOT, CMS Combine, cppyy, BioDynaMo, xeus, conda-forge. Only pull requests authored by people on compiler-research.org/team count; LLVM before Oct 2023 counted as commits.

Attribution: current team all years; alumni up to their program year; ROOT work by people whose day job is ROOT only when it is about the interpreter, interop or AD.

Contributors: distinct people with merged work. Community: authors not on the team page.

Not used: lines of code (dominated by code migrations and imports).

New contributors: first merged work in our data, which starts in 2021. Cling: root-project/cling counted in full.

Checked: Clad merged pull requests (501) and Vassil Vassilev's LLVM pull requests (15) match GitHub search totals exactly.

Backup: Talks, training and outreach

We organized

28th EuroAD Workshop (Dec 2025)

CompilerResearchCon 2026

NVIDIA at CERN meeting (Jul 2026)

Conference and workshop talks (12)

ROOT Users Workshop, Valencia (Nov 2025): Vassil Vassilev; Aaron Jomy and Vipul Cariappa; Salvador de la Torre Gonzalez (CARTopiaX)

28th EuroAD Workshop (Dec 2025): Aaron Jomy (teaching AD); Petro Zarytskyi (Clad, object-oriented C++)

IRIS-HEP Differentiable Analysis Blueprint, CERN (Mar): Jonas Rembser and Vassil Vassilev, Differentiable Programming in C++ and ROOT with Clad

using std::cpp 2026, Madrid (Mar): Aaron Jomy and Vipul Cariappa, Compiler as a Service: C++ Goes Live

EuroLLVM 2026 (Apr): Aaron Jomy, CppInterOp: Interactive C++ as a Service

CHEP 2026 (May): Aaron Jomy, ROOT's new Python interfaces based on CppInterOp

ErUM-Data workshop on ML for Universe and Matter, Aachen (Jun-Jul): Vassil Vassilev (with Uwe Naumann, RWTH Aachen) and Jonas Rembser, Automatic Differentiation

MODE 2026, Crete (Sep): Vassil Vassilev with co-author Jonas Rembser, AD in CMS Combine: From Integration to Physics-Grade Fits; Trust, then scale: checkable gradients

Training (4)

EuroLLVM 2026 (Apr): Aaron Jomy, hands-on Using Clang as a library

ISC 2026, Hamburg (Jun): Aaron Jomy and Jonas Rembser, Accelerated Python tutorial on NVIDIA's Accelerated Computing Hub

CoDaS-HEP 2026 school, Princeton (Jul): Vassil Vassilev, Introduction to Automatic Differentiation and Differentiable Programming; Performance Tuning of Scientific Software

Upcoming: SC26 (16 Nov), full-day Accelerating and Scaling Python for HPC

Backup: Projects since CRCon25

Completed, with merged work (11)

Supporting the Thrust API in Clad: Abdelrhman Elrawy (GSoC)
Compiler-driven LLM training with Clad: Rohan Timmaraju (GSoC)
Layered tape with prefetching in Clad: Aditi Milind Joshi (GSoC)
Activity analysis for CUDA kernels in Clad: Maksym Andriichuk (IRIS-HEP)
Automatic differentiation of OpenMP programs: Jiayang Li (GSoC)
Debugging support for xeus-cpp: Abhinav Kumar (GSoC)
RAMTools, ROOT for genomic data: Aditya Pandey (GSoC)
Advanced symbol resolution for clang-repl: Sahil Patidar (GSoC)
CARTopiaX extended to a wet-lab experiment: Salvador de la Torre Gonzalez (GSoC)
RAMTools improvements: Georgi Haralanov (high school)
Teaching materials for C++ and CUDA with xeus-cpp: Hristiyan Shterev (high school)

In progress

Clang: C++26 features, diagnostics, clang-tidy triage, modules, performance (C++ Alliance fellows)
CppInterOp memory ownership and thread safety: Kerem Şahin (GSoC)
Clad GPU infrastructure: Vedant Goyal (GSoC)
Clad in LibTorch: Kacent Huang (GSoC)
Differentiable rendering in Clad: Abdelrhman Elrawy (GSoC)
Clang error recovery: Sahil Patidar
On-demand parsing in Clang: Vipul Cariappa
GeneROOT infrastructure: Jeffrey Zhang
GSoC 2026 final evaluations come after CRCon26

Not completed or not merged: 5 (one GSoC 2025 project, four high-school projects)

Backup: People beyond programs

6 high-school interns

10th graders from the Math High School in Plovdiv

All 6 wrote about their projects on the blog (9 posts); 2 with merged code so far

The aim: from curiosity to real upstream contributions

Outside our programs

31 people outside any program got 92 pull requests merged

From CERN and QuantStack engineers to first-time contributors

Volunteers on the team: Matthew Barton, Jeffrey Zhang



Backup: Who is the team?

Team = everyone on compiler-research.org/team, current and alumni: 75 people, 23 current

Leads and associated

2 leads and 6 associated members at partner organizations

Program contributors

GSoC, C++ Alliance fellows, IRIS-HEP fellows, CERN summer students, Season of Docs writers, high-school interns

Volunteers and staff

Open source contributors, interns, contractors and research staff

Community

Anyone whose work was merged into our repositories: 31 people since CRCon25