



std::core_dump

The Newsletter of Boost and the C++ Alliance



Issue 3
April 2026

In this Issue

Boost.Decimal Debuts in Boost 1.91

by Matt Borland

Boost.Graph Modernization Workshop

by Arnaud Becheler

Multi Review Verdict – Accepted Conditionally

by Matt Borland

The Road to 'Import Boost': a Library Developer's Journey into C++20 Modules

by Rubén Pérez Hidalgo

Q1 2026 Systems & CI Update

by Sam Darwin

Monthly Metrics

Farewell to Wowbagger - End of an Era

by René Ferdinand Rivera Morell

The Mathematical Mind of a C++ Programmer

by Joaquín M López Muñoz

WG21 Croydon – March 2026

by Harry Bott



Boost.Decimal Debuts in Boost 1.91

by Matt Borland

Boost 1.91, shipping April 22, marks the first official release of Boost.Decimal, a header-only, dependency-free C++14 implementation of IEEE 754 and ISO/IEC DTR 24733 decimal floating-point arithmetic, authored by Matt Borland and Chris Kormanyos.

Binary floating point can be a poor fit when values need to round-trip exactly the way humans write them, as is seen in financial calculations. Boost.Decimal provides developers a set of drop-in types (`decimal32_t`, `decimal64_t`, `decimal128_t`) that behave like built-in floating point types, but store and compute using base-10 arithmetic instead of base-2. The library is comprehensive; it ships its own implementations of many standard library headers (`<cmath>`, `<charconv>`), and support for external libraries like `{fmt}`. Boost.Decimal is tested on a variety of operating systems and architectures to ensure that it is performant across diverse operating environments.

The road to acceptance included not one, but two formal reviews. We made substantial feedback-driven improvements due to these reviews. If you use this library and find it useful let us know. We look forward to continued feedback as the library is more broadly used and distributed.

Links:

Repository: <https://github.com/boostorg/decimal>

Documentation: <https://www.boost.org/doc/libs/develop/libs/decimal/doc/html/overview.html>

Cplusplus Slack - #boost or #boost-decimal





Boost.Graph Modernization Workshop – May 6th, Paris

by Arnaud Becheler

As part of an ongoing effort with the C++ Alliance to modernize Boost.Graph, we are organizing a small focused workshop on May 6th, 2026 in Paris, with full remote participation available.

The goal is to bring together graph algorithm researchers, C++ library contributors, and industrial users for an honest conversation about what Boost.Graph should look like in 2026 and beyond. The day will combine short lightning talks (use cases, tools, pain points) with structured discussions, ending with a concrete prioritized roadmap.

If you work with graphs, have unmet needs, or simply have opinions on BGL's future, you are warmly invited. Even if you cannot attend, sharing your use cases or algorithm wishlist in the GitHub discussion will directly feed into the roadmap.

Contact: arnaud.becheler@gmail.com

Multi Review Verdict – Accepted Conditionally

by Matt Borland

Boost.Multi is a C++ library by Alfredo Correa that provides manipulation and access of data in multidimensional arrays for both CPU and GPU memory.

For transparency, the final tally of ACCEPT/CONDITIONAL ACCEPT/REJECT, from both private and public reviews, was 1/5/2. Both rejections and nearly all conditional acceptance conditions revolved around the state of the documentation and naming. I have consolidated these conditions and related discussion on the GitHub issue tracker, rather than reiterating them here. The total number of new issues from the review period is 25. Rainer and Peter Turcan both brought up bugs or additional features for the website. I am working to address those separately with the web devs.

This leads us to the singular condition of acceptance being to address the documentation related issues. Since this is a rather subjective requirement, we will be running a mini review of the library documentation in a few weeks. Several of those who conditionally accepted or rejected said they would be willing to look at the library again. This mini review period provides them a good opportunity to do so. I think this second review will give the best possible product for the initial release of Multi in Boost. Stay tuned for those dates.

Thank you again for your time and participation in this review. The library is certainly better for it.

The Road to ‘Import Boost’: a Library Developer’s Journey into C++20 Modules

by Rubén Pérez Hidalgo, presented at using std::cpp 2026

C++20 modules have been in the standard for more than 5 years now. They promise a fundamental change to how we write C++, but adoption hasn't been as widespread as many expected. This talk takes a deep dive into the practical aspects of C++20 modules, exploring the reality of the ecosystem as it stands today.

The session begins by covering how to write module-native applications, introducing all the required core concepts. It then examines potential build time improvements in the context of a real-world application that uses Boost – spoiler: having `import boost` would save the day! Finally, it explores the options available to library authors for adding C++20 module support, including a prototype currently being developed at Boost.

No prior experience with C++20 modules is required to follow along.

Slides: <https://github.com/anarthal/usingstdcpp-2026/blob/master/slides.pdf>

Video: <https://www.youtube.com/watch?v=hD9JHkt7e2Y>

Q1 2026 Systems & CI Update

by Sam Darwin

Code Coverage Reports – The team continued developing codecov-style coverage reports that run in GitHub Actions and are hosted on GitHub Pages. The upstream GCOVR project has taken interest in the custom templates, and work is underway to merge them into the main GCOVR repository for all users. Julio Estrada led the template development, which now includes improved source file scrolling, function listings on every page, optional branch coverage, blue and green themes, and new forward/back navigation buttons.

Server Hosting – The legacy Rackspace servers (10–15 years in service) were decommissioned. A replacement server, wowbagger2, was built to host original boost.org, running on a GCP Compute instance at roughly 5% of the old monthly cost. New server instances were also launched for corosio.org and paperflow.

Fil-C – Work is progressing with Tom Kent to add Fil-C (a memory-safe C compiler) to the Boost regression matrix. A container image has been built and the compiler is successfully building – expect it online soon.

Release Tools – CircleCI build times were reduced significantly by increasing parallelism (4→8 processes), upgrading instance sizes, and swapping in parallelized compression tools (lzip2, pigz). A Sphinx pip package upgrade is staged and ready to go live after the 1.91.0 release.

Doc Previews & Builds – Fixed the Antora "Edit this Page" feature for Boost's git-submodule structure (credit to Ruben Perez). Rolled out the antora-downloads-extension across numerous repos and refactored build_docs scripts to use standard Gemfile and requirements.txt formats.

Boost Website (V3) – Supported the onboarding of consulting firm Metalab for the next iteration of boost.org. New tooling includes a QA deployment script for testing any open pull request by commit SHA, and an image upload script for storing Bob Ostrom's Boost cartoons in S3.

Jenkins – Optimized disk usage by relocating the Node.js installation to a shared /opt/nvm directory, eliminating redundant 1GB node_modules installs across dozens of doc-preview jobs. Dmitry contributed diff-reports that now appear in LCOV-configured pull requests.

GHA & Drone – Completed a major Terraform upgrade (closing a two-year gap), deployed runners for Joaquin's boost_hub_benchmarks with VS2026 and macOS 26.3, and launched new macOS 26 and FreeBSD 15.0 Drone runners



Jan website traffic

49k
uniques

Total repositories using
Boost libraries

(repos with 10+ stars)

12,615

X followers

2,487



Space Communications Cognitive Engine by
Penn State & NASA (in progress)
https://www.boost.org/testimonials/sven_bilen_and_timothy_hackett/



Farewell to Wowbagger - End of an Era for boost.org

by René Ferdinand Rivera Morell

On behalf of the Fiscal Sponsorship Committee, I'm writing today with a mix of gratitude and nostalgia to share some news: after 15 years of faithful, but rocky, service, wowbagger - the server that hosted boost.org, lists.boost.org, and subversion - is being mothballed. For many of us, it was simply there - one of those invisible pillars that kept our community connected and our work accessible to the world. It deserves a proper send-off.

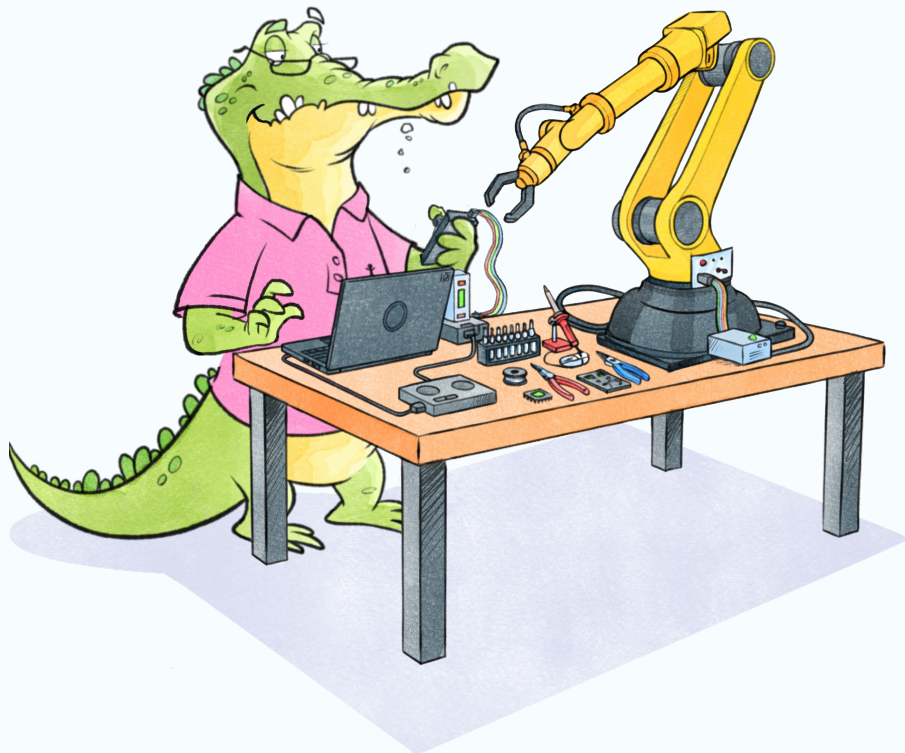
What happened: Rackspace recently notified customers that pricing on their legacy cloud platform is increasing significantly - effectively doubling - as they wind down the previous generation of cloud servers in favor of their next-generation platform. Migration is not automatic. After consulting with Marshall and others, we made the decision to terminate our Rackspace instances rather than absorb the increased costs or undertake a manual migration to a platform we no longer need.

What this means: We (i.e. Sam Darwin) have stopped wowbagger for now. File-level backups and disk snapshots are in hand. After another week, the server will be deleted. This is purely a housekeeping matter - boost.org and lists.boost.org are already running on new, upgraded hardware and have been for some time. The old machine had been kept around as a reference for comparing web pages between the old and new sites, but the Rackspace pricing change simply accelerates a retirement that was already overdue.

Nothing changes for us. The website and mailing lists continue to operate normally on their new home.

Still, we wanted to take a moment to honor wowbagger's long run. 15 years is a remarkable tenure for any server, and this one carried the public face of Boost for all of that time. To everyone who set it up, maintained it, patched it at odd hours, and kept it humming along - thank you. Your work mattered more than most people ever realized.

Rest well, wowbagger. You've earned it.





The Mathematical Mind of a C++ Programmer

by Joaquín M López Muñoz,
presented at using std::cpp 2026

Many people hate math, some programmers do too. We hold that this hate stems from a reductionist view of mathematics as merely rule application: number crunching, equation solving and the like. But programming is more akin to mathematical creation, and writing a well designed, neat program can be as exhilarating as devising a new little math theory.

In this talk we investigated how the mathematical mind approaches the world, and how developing a mathematical inclination can help you be a better C++ programmer.

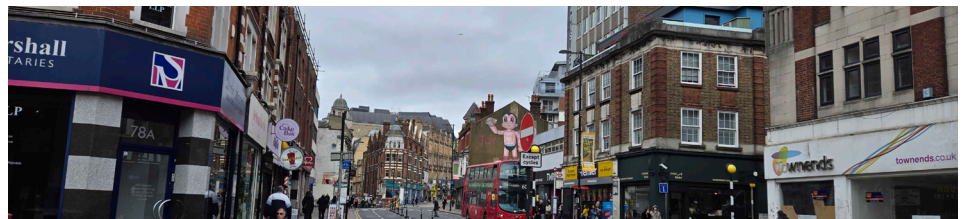
Presentation: <https://github.com/joaquintides/usingstdcpp2026>

CppCon 2026 Call for Presenters

Consider doing a talk at CppCon 2026! We'd be happy to help you submit a proposal. Whether it's a technical deep-dive, project update, or lessons learned – it's a great opportunity to connect with the community.

Reach out to Mark Cooper and we'll start the process.

Contact: mark@cppalliance.org



WG21 Croydon – March 2026

by Harry Bott

Last month several of us represented the C++ Alliance at the WG21 ISO C++ committee meeting in Croydon, from March 23 to 28. A personal highlight for me was meeting Vinnie Falco in person for the first time after working together remotely. Vinnie was there alongside Mungo Gill and Matheus Izvekov, with our assistant Emma supporting throughout the week as well. Between sessions, hallway conversations, and the usual shared meals, the week had the real WG21 rhythm: serious technical work, constructive discussion, good people, and clear momentum behind the language.

The formal work spanned plenary, LEWG, and smaller groups, with strong engagement around networking, senders and coroutines, profiles and safety, and proposal quality. That is exactly the mix of today's standard work and next-step thinking that makes these meetings worthwhile. Saturday's closing plenary marked an important milestone, with national body votes moving C++26 one step closer to international ballot. It reflects years of work from hundreds of people. We also have new Alliance members in the pipeline, and we will be announcing them soon. On behalf of everyone who attended, thank you to the committee and the hosts for a productive week. We are already looking ahead to the next meeting in Brno.