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

Interactive C++: cling and clang-repl

Garima Singh

06.04.2023

Status. Cling

- No news here: We need to fix 6 tests in Cling

Status. Clang-Repl

- Incremental Input (RFC)
  - D143142 — Enable Lexer to grow its buffer
  - D143144 — Add TryGrowLexerBuffer/SourceFileGrower
  - D143148 — Add basic multiline input support
- Value Handling (RFC)
  - D141215 — Introduce Value and implement pretty printing
  - D146389 — Initial interactive CUDA support for clang-repl

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

Status. InterOp

- Works with both Cling and ClangRepl now.
- libInterOp-based cppyy: pass 130/498 tests.

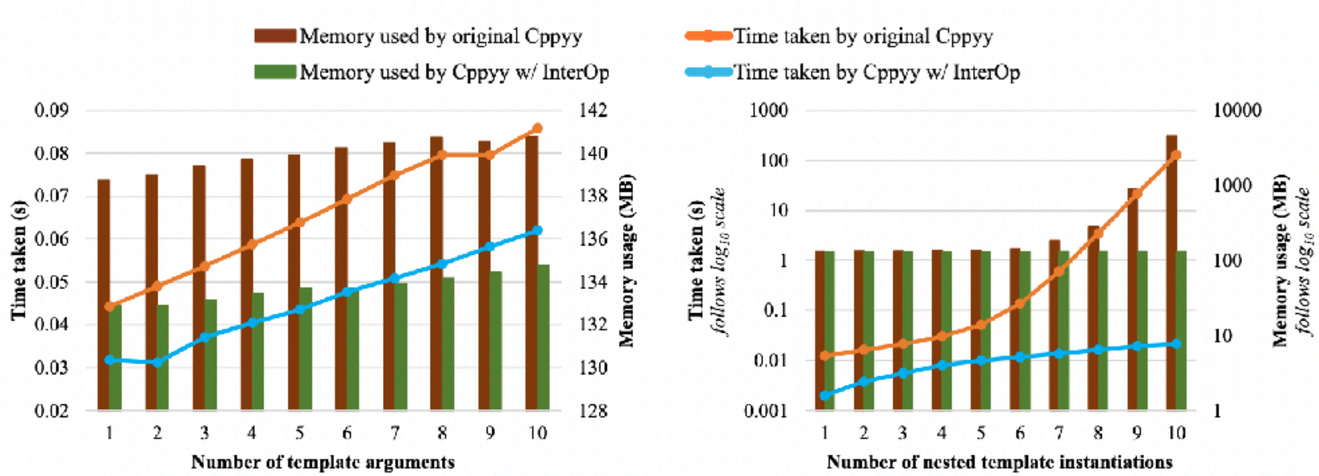

Figure 3. Time taken and memory used during class template instantiation. On the left, we compare template instantiations with `std::tuple<double, double, ...>` where the number of template instantiations done by the C++ interpreter increases with the number of template arguments. On the right, we compare instantiating nested templates, for example, `std::vector<...<std::vector<double> >`, where cppyy has to instantiate each nesting individually from the innermost to the outermost class template. These are common features of high-performance, templated numerics libraries that utilize template expressions.

Status. Clad

- Several promising GSoC candidates applied. We might get good contributions soon.

Status. Xeus-Clang-Repl

- Started a new project Xeus-Cpp in collaboration with QuantStack. The idea is to replace xeus-clang-repl and xeus-cling. Xeus-Cpp must be able to work with upstream llvm and clang. It must also support running in a web browser through WebAssembly.

Upstreaming Patches

- Spreadsheet tracking the progress here.
- Total amount of upstreamed cling patches 26(26+0) out of 52 upstreamable.

CaaS Open Projects

- Open projects are tracked in our open projects page.

Next Meetings

- Monthly Meeting — 4th 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!