GSOC 2022 Final Report

- **Name**: Matheus Izvekov
- **Email**: mizvekov@gmail.com
- **Github Username**: mizvekov
- **Organization**: LLVM

- **Project Title**: Template Specialization Resugaring for Clang

About Clang

Clang is an open-source compiler targeting languages in the C family, part of the LLVM project. It has excellent code generation quality, thanks to it's architecture and targetting of LLVM IR. It also has a powerful AST, capable of representing the source code as written better than most compilers, making it specially suitable for this project.

Project Description

Template specialization resugaring aims to overcome a jarring limitation in the compiler's support of templates, where the sugar of template arguments is not preserved. This leads to well known problems in diagnostics related to templates, where familiar names provided by used aliases are not presented to the user. This project implements a new pass which adds the lost sugar back upon access to the templates, with much reduced overhead compared to an approach where templates are simply instantiated for all types as written.

Approach and workflow

More details given in slides: Initial Roadmap Final Roadmap

Goals

- General improvements to type sugar retention.
  - Retain type sugar in template argument deduction.
  - Retain type sugar in 'auto' types.
  - AST improvements to retain lack of elaboration.
  - AST improvements to the representation of expanded packs.
  - Calculate the common sugared type for deduction and most binary operations.
  - AST improvements for builtin templates
  - AST improvements for the representation of template substitutions.
  - Retain type sugar in template argument checking.
- Improve template instantiation for non-uniqued specializations.
  - Support instantiations with non-canonical arguments.
  - Improve AST so that markers are not left behind for final substitutions.

  - Preserve type sugar in NTTP instantiation.
  - Preserve type sugar in default argument instantiation.
  - Preserve type sugar in instantiations of concepts.
  - Preserve type sugar in instantiations of type aliases.
- Template Specialization Resugaring
  - Implement resugaring pass.
  - Resugar accesses to general type declarations.
  - Resugar accesses to type alias patterns.
  - Resugar expressions.

Landed Patches

- [clang] retain type sugar in auto / template argument deduction
- [clang] Implement ElaboratedType sugaring for types written bare
- Clang: fix AST representation of expanded template arguments.
- [clang] template / auto deduction deduces common sugar
- [clang] use getCommonSugar in an assortment of places
- [clang] extend getCommonSugaredType to merge sugar nodes
- [clang] Fixes how we represent / emulate builtin templates
- [clang] AST: SubstTemplateTypeParmType support for non-canonical underlying type
- [clang] Implement divergence for TypedefType and UsingType
- [clang] Track the templated entity in type substitution
- [clang] Changes to produce sugared converted template arguments
- [clang] Perform sugared substitution of builtin templates
- [clang] Implement sugared substitution changes to infrastructure
- [clang] Instantiate NTTPs and template default arguments with sugar
- [clang] Instantiate concepts with sugared template arguments
- [clang] fix profiling of template arguments of template and declaration kind
- [clang] Improve diagnostics for expansion length mismatch
- [clang] Fix missing template arguments in AST of access to member variable template
- [clang] Improve error recovery for pack expansion of expressions
- [clang] Fix handling of unexpanded packs in template argument expressions
- [clang] fix missing initialization of original number of expansions
- NFC: [clang] Template argument cleanups.
- [clang] AST/Print: honor AlwaysIncludeTypeForTemplateArgument policy
- [clang] Implement setting crash_diagnostics_dir through env variable

Ongoing Patches

- [clang] Instantiate alias templates with sugar
- [clang] store sugared converted arguments on TemplateSpecializationType
- [clang] Template Specialization Resugaring - TypeDecl
- [clang] Template Specialization Resugaring - Template Type Alias
- [clang] Store in exprs the deduced arguments for function calls.

- [clang] Misc type sugar preservation improvements
- [clang] Template Specialization Resugaring - Expressions
- [clang] ASTImporter: Fix importing of va_list types and declarations
- [clang] Fix missing diagnostic of declaration use when accessing TypeDecls through typename access

Communication and Work Management

- Text communication was available with all mentors. We had clear communication, with little delay. Feedback provided was thoughtful and much appreciated.
- Code review was mostly done through outside contributors. This was an area where we often became blocked in the early part of the project, due to the nature of the project and lack of experts available to review.
- We had weekly meetings where we checked on each others progress, and discussed and decided on ways forward through many blockers.
- In some of the meetings, we gave more detailed presentations of our overall project and roadmap.
- I had complete control of my time available for the project, and planned and kept track of the goals needed to complete the project.

Future Scope

TODOs

- A few more performance enhancements for the resugaring pass.
- We need to investigate a better performant way to preserve converted template arguments on TemplateSpecializationTypes.
- Review on the main resugaring patch needs to progress.
- A couple of regressions need to be fixed.

Long Term Goals

- Resugaring coverage improvements:
  - Base classes.
  - Nested accesses in expressions.
- Improvements to the diagnostics of template instantiations themselves. They can, with a few assumptions, be instantiated fully sugared while still being uniqued, provided resugaring can produce the correct results upon accesses to these specializations.