GSoC 2022 Project Report

Add Initial Integration of Clad with Enzyme

Student: Manish Kausik H

Organisation: CERN-HSF

Mentors: Vassil Vassilev, David Lange, William Moses

Project repository: Clad

Project proposal: Proposal link

Overview of the Project

Project Abstract

Clad is an open source plugin to the Clang compiler that detects calls to differentiate a defined function from the parsed Abstract Syntax Tree(AST), generates code that is the derivative the function and modifies the AST to insert the generated code. Clad works using the concept of Automatic Differentiation(AD), which in mathematics and computer algebra is a set of techniques to numerically evaluate the derivative of a function specified by a computer program. While Clad works in the frontend of the compilation process, Enzyme, which is an LLVM based AD plugin, works in the backend, where it takes in code in LLVM IR form and then differentiates the code. This Project aims to integrate Enzyme within Clad, and give a Clad user the option of selecting Enzyme for Automatic Differentiation, based on his/her needs. This will give the user the same User Interface as Clad for writing his/her code, but the option of using Enzyme as the backend with all its optimisations to calculate the derivative of the requested function.

Project Deliverables

This project proposes to add the following features to Clad:

1. Define the Enzyme configuration requirements and enable Clad to communicate efficiently with Enzyme.
2. Enable Clad to use Enzyme's Reverse Mode AD feature when requested by the User.
3. Document the above features and write unit tests for them.

Contributions

The major PRs I contributed in the GSoC period are mentioned here. All of them have been merged.

PR	Description
#460	Detect Request for use of Enzyme as Backend by user
#466	Generate code for Enzyme autodiff for functions with Pointer/array arguments
#486	Add Support for Differentiating functions with both pointer/array type and primitive type parameters with Enzyme
#491	Benchmark Clad and Enzyme
#492	Treat Request to use Enzyme for Differentiating a Function to be different from a Clad differentiation
#494	Add Documentation for Integration of Enzyme with Clad
#495	Add tests to verify enzyme results with clad

To see all the PRs contributed to Clad by me checkout thislink

Results

Currently Support for using Enzyme for Reverse Mode AD has been enabled within Clad. Here is an example that demonstrates how an user of Clad can request for using Enzyme as the backend for AD.

```
#include "clad/Differentiator/Differentiator.h"

double foo(double* arr) { return arr[0] * arr[1]; }

int main(){
    auto grad = clad::gradient<clad::opts::use_enzyme>(f1);
    double v[2] = {3, 4};
    double g[2] = {0};
    grad.execute(v, g);
    printf("d_x = %.2f, d_y = %.2f\n", g[0], g[1]);
}
```

Thus, the calling convention is to use `clad::gradient<clad::opts::use_enzyme>(...)` instead of the usual calling convention, `clad::gradient(...)`.

To read more about Enzyme and Clad Integration please visithere

Work Left

Forward Mode AD is still under development in Enzyme. When a stable release for forward mode AD is available in Enzyme, we can add support for using Enzyme as backend for forward mode AD within Clad.

Acknowledgements

I thank my mentors Vassil Vassilev, David Lange and William Moses for helping me in making my project successful. I am especially grateful to Vassil for helping me when I faced bottlenecks and guiding me at crucial moments in the project. I am also thankful to my fellow students Parth Arora, Garima Singh and Baidyanath Kundu for helping me with the Clad codebase when I was stuck at understanding it.