

# OpenACC V&V Suite for LLVM

## Project Goal

The primary goal of this project is to facilitate the use of the OpenACC V&V Suite [0] for testing the behavior of LLVM's current and future OpenACC support. We anticipate this project will immediately benefit the development of Clacc (OpenACC support for C/C++ in Clang) and Flacc (OpenACC support for Fortran in Flang). Moreover, we anticipate it will establish feedback from Clacc and Flacc to the OpenACC V&V Suite team to advance testing for the latest OpenACC specification versions, which are important to ensure application portability among OpenACC implementations.

## Testing Role

Clacc and Flacc are developing their own test suites to be contributed to upstream LLVM. Unlike the OpenACC V&V Suite, these test suites are not meant to be general to all OpenACC implementations. For example, Clacc's test suites exercise Clacc-specific command-line options and source-to-source capabilities, which are beyond the scope of the OpenACC specification. The major value of the OpenACC V&V Suite would be an objective, third-party assessment of LLVM's conformance to the OpenACC specification that can be compared with other OpenACC implementations.

## Tasks

There are several potential tasks, depending on how much time a contributor has:

1. Integrate the OpenACC V&V Suite into upstream LLVM. Part of the effort here is to engage with the Clacc, Flacc, OpenACC, and upstream LLVM communities to determine the most useful form of that integration. See below for initial ideas.
2. Automate detection of test result changes. We need a way to record known failures for each of Clacc and Flacc so that only new failures (or new passes) are reported. One challenge is that, until Clacc is contributed to upstream LLVM, Clacc's record of known failures might have to be kept in Clacc's repo instead of upstream. Another challenge is coordinating the record with versions of the OpenACC V&V Suite.
3. Catalog failures and file issues. Is it a Clacc, Flacc, or OpenACC V&V Suite issue? Is it an unimplemented feature or a bug in an implemented feature? How severe is the issue?
4. Start fixing issues.

# How to Integrate

LLVM has multiple test suites [1].

First, tests within LLVM's main github repo [2] are usually described as unit tests or regression tests. These are the types of tests that Clacc and Flacc are developing already, as discussed above.

Second, LLVM has a "test-suite" [3], which is maintained in a separate repo [4]. This is where stand-alone programs and external test/benchmark suites are typically integrated. This is likely where the OpenACC V&V Suite should be integrated.

Within LLVM's "test-suite", some external suites are not actually added directly to the repo. Instead, hooks are added to run those suites when they are available on the user's system. For example, some SPEC suites are integrated in this manner [5]. We need to consider whether this is the right strategy for the OpenACC V&V Suite. Considerations include licensing and version control.

## Who to Contact

Clacc: Joel E. Denny

Flacc: Valentin Clement

OpenACC V&V Suite: Sunita Chandrasekaran

While Joel and Valentin are upstream LLVM developers, some conversations need to involve a broader portion of the LLVM community or may cover topics with which Joel and Valentin have less experience. There are many people who would likely be willing to answer general questions.

## References

[0] <https://github.com/OpenACCUserGroup/OpenACCV-V>

[1] <https://llvm.org/docs/TestingGuide.html>

[2] <https://github.com/llvm/llvm-project/>

[3] <https://llvm.org/docs/TestSuiteGuide.html>

[4] <https://github.com/llvm/llvm-test-suite.git>

[5] <https://github.com/llvm/llvm-test-suite/tree/main/External/SPEC>