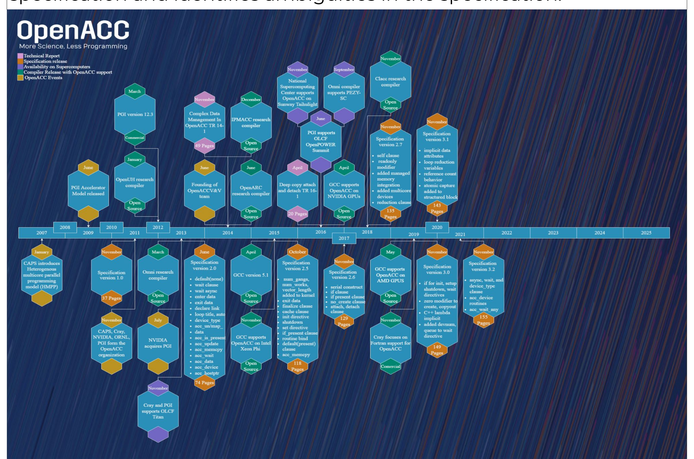
Analysis of Validating and Verifying OpenACC Compilers 3.0 and Above

Aaron Jarmusch, Aaron Liu, Christian Munley, Vaidhyanathan Ravichandran, Daniel Horta, Joel Denny, Kyle Friedline, Sunita Chandrasekaran (jarmusch, olympus, chrismun, dchorta, vaidhy, schandra) @udel.edu

Overview of Model & Testsuite

OpenACC
- Directive-based programming model targeting parallel computing.
- Specification is constantly evolving to accommodate the needs of application developers.

OpenACC Validation and Verification testsuites is a Third Party Validation testsuite that: evaluates compilers' compliance with the specification and identifies ambiguities in the specification.



Contribution to Community

- This Work Contributes:
  1. Provides a Testing **Infrastructure** for C/C++ and Fortran tests
  2. Identify and **Report** Compiler bugs and runtime errors
  3. Evaluate different compilers' implementations for its **conformance** to the OpenACC specification

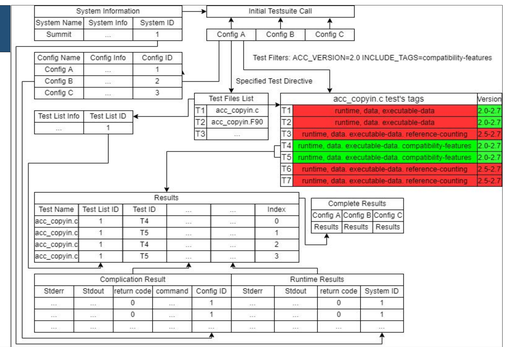
Challenges

1. Code compilation and targeting device differ w/ Compilers - Systems
2. Some features are difficult to test based on the definition
3. Certain features can only be tested by stressing to failure
4. Few tests based on features yet to be implemented in compilers
5. Example of a challenging test: Init Directive w/ If Clause

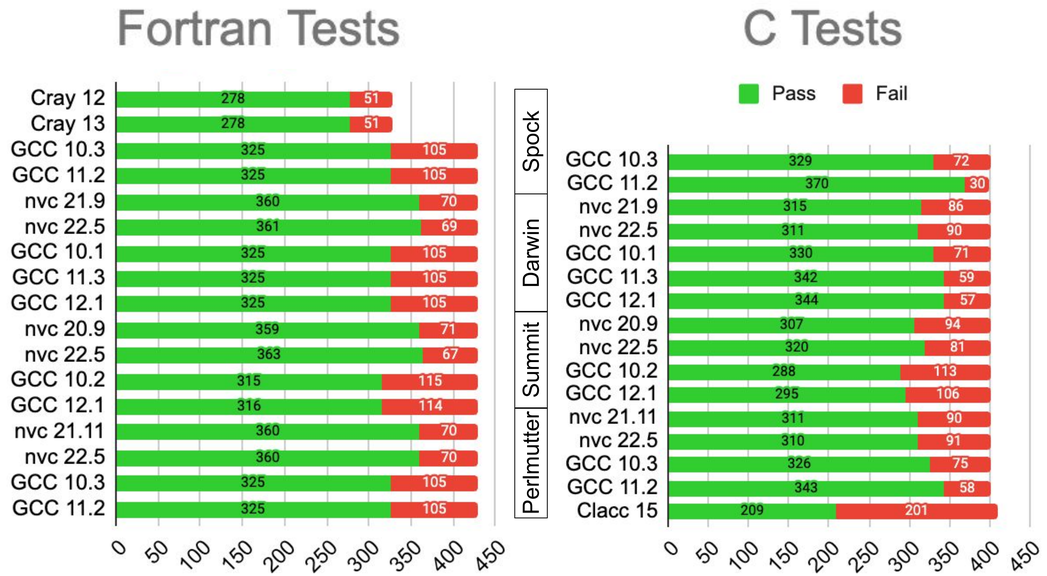
```
int err = 0;
srand(SEED);

int device_num = acc_get_device_num(acc_get_device_type());
#pragma acc init if(device_num == device_num)

return err;
```

Infrastructure

- Made to run the tests with **customization** and **simplicity**, it can be tailored to any system through the configuration file.
- The most important features to the average user of the infrastructure are compiler and compiler flag selection, output format specification, conditional compilation, and hang-time limits.
- A comprehensive list of all of the options can be found here https://github.com/OpenACCUserGroup/OpenACCV-V



Results



Spock	DARWIN	Summit	Perlmutter
AMD MI100	NVIDIA T4	NVIDIA V100	NVIDIA A100

LLVM Integration

- Using the OpenACC V&V testsuite would be an objective, third-party assessment of **LLVM's (CLACC)** conformance to the OpenACC specification that can be compared with other OpenACC implementations
- Clacc compiler translates OpenACC to OpenMP to build upon the OpenMP support being developed for Clang and LLVM. A CMakeList file has been included in this project which is the entry point to our llvm-test-suite. It generates the Makefile which allows users to compile, run, and report test results. LLVM OpenACC V&V Intergration found here: https://github.com/llvm-doe-org/llvm-test-suite/tree/llvm.org/main/External/openacc_vv

Analysis of Results

- Compilers have different features implemented
- **Improvements** occur with newer versions of compilers
- Results are primarily consistent across architectures
- Provides a metric to measure future versions against
- Some compilers translate OpenACC to OpenMP, where some features are not equivalent between both models

Conclusion - Future Work

- The purpose is to **validate and verify** the implementations outlined by the **OpenACC specification** for **all systems**. As seen within the result section, many of the features have failed the tests: serial loop, atomic capture, routine bind functions with lambda functions, and the combination of the data clause copyin and copyout, to list a few.
- The C++ tests are not fully supported within the infrastructure, and there are plans to comb through to resolve this issue.
- Our future goals are to continue developing tests for new features and to cover more edge cases. Using examples, we will also create an **example guide** that will give a beginner-friendly breakdown of how to implement OpenACC.

Acknowledgments/ References

This material is based upon work supported by the National Science Foundation (NSF) under grant no. 1814609. We would also like to thank immensely OpenACC for their tremendous support towards GPUs, technical support and funding.

1. OpenACC Specifications. https://www.openacc.org/specification
2. OpenACC 1.0," *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*, 2014, pp. 1407-1416, doi: 10.1109/IPDPSW.2014.158.
3. Friedline K., Chandrasekaran S., Lopez M.G., Hernandez O. (2017) OpenACC 2.5 Validation Testsuite Targeting Multiple Architectures. In: Kunkel J., Yokota R., Taufer M., Shalf J. (eds) High Performance Computing. ISC High Performance 2017. Lecture Notes in Computer Science, vol 10524. Springer, Cham. https://doi.org/10.1007/978-3-319-67630-2_3