

# State of LLVM Flang Development

Leandro Lupori  
Carlos Seo



# What is Flang?

- Flang is LLVM's Fortran frontend
- LLVM Flang != Classic Flang
- Ground-up implementation written in modern C++
- Generates MLIR from Fortran code:
  - FIR (Fortran IR), plus other MLIR dialects
- MLIR is then lowered to LLVM IR and then to machine code

# Fortran support overview

- Fortran 95 – practically complete
- Fortran 2003 – mostly supported
  - Polymorphic types: complete
  - Procedure pointers: initial support
  - Interoperability with C: improved
- Fortran 2008-2018
  - Some features supported
  - Can be parsed by the Frontend

# GPU/target offloading

- CUDA Fortran support was added recently
- Improved support for offloading sections of a program to GPU/other accelerators, using OpenMP/OpenACC

# OpenMP

- Support for OpenMP 1.1 is practically complete
- Working towards 2.5 completion
- Some features of later standards are already supported
- Now works with SPECspeed® 2017

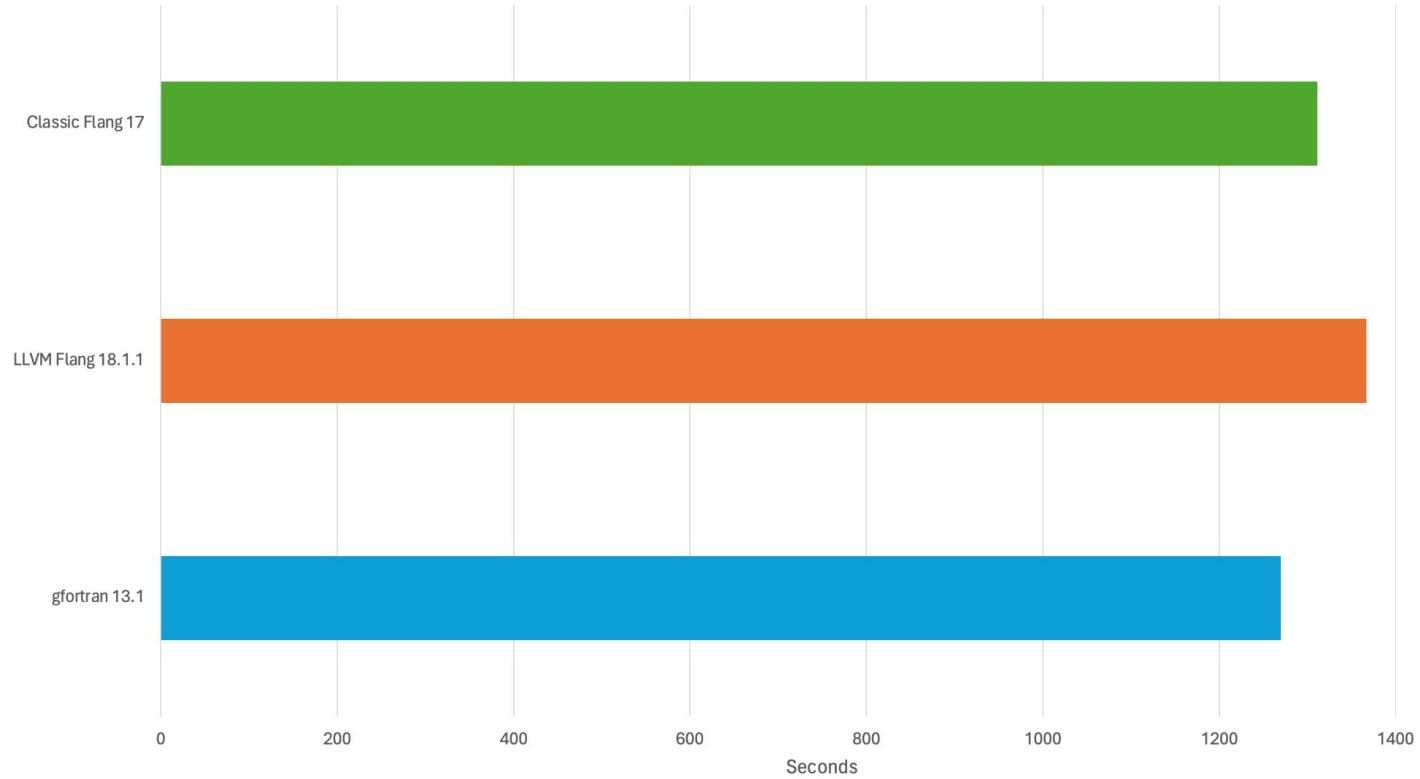
# Testsuites

- GFortran tests added to LLVM testsuite
  - Around 80% of the tests pass with Flang (excluding unsupported GFortran specific features)
- Fujitsu and IBM are working to upstream their Fortran tests too
- Testsuites are driving many bug fixes and improvements

# Performance highlights

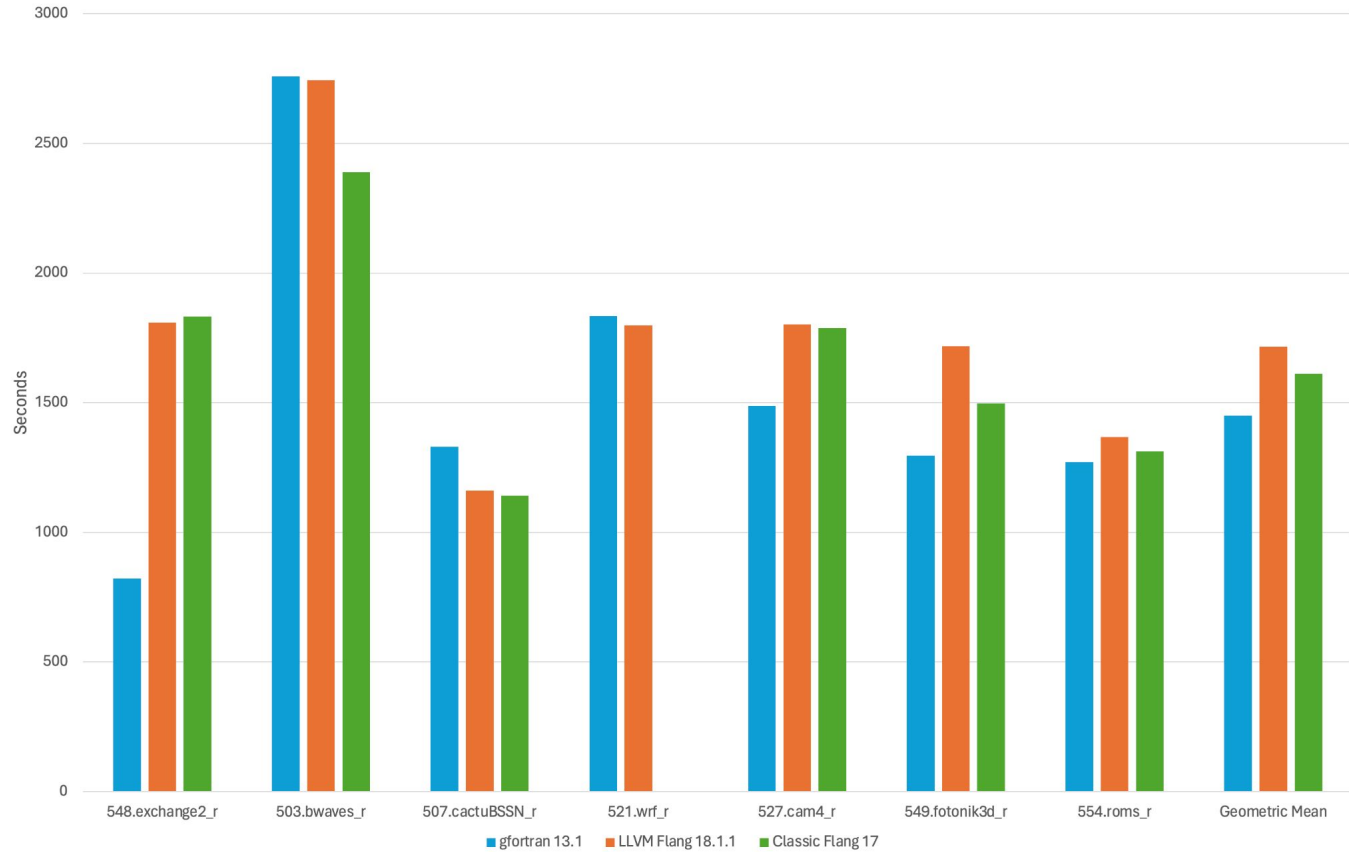
- TSVC (Test Suite for Vectorizing Compilers)
  - Performance analysis and improvements of vectorization
- TBAA (Type Based Alias Analysis)
- HLFIR (High Level FIR) work completed
  - More opportunities for optimization
  - Better debugging
- SPEC CPU® 2017
  - LoopVersioning - improved performance of 554.roms\_r

### SPEC® CPU 2017 Rate Results - 554.roms\_r





### SPEC® CPU 2017 Rate Results - Fortran Benchmarks



# Performance of LLVM Flang 18.1.1

- SPEC CPU® 2017 Rate - Fortran Benchmarks (AArch64)
  - < 7% slower than Classic Flang
  - < 20% slower than GFortran
  - ~37% faster than LLVM Flang 16.0.6

[\[Comparing LLVM Flang with other Fortran compilers\]](#)

# What is missing?

- Not ready yet for production usage
  - [Rename](#) flang-new to flang
    - Depends on the state of the gfortran testsuite
  - Some missing features
    - i.e. procedure pointers support
  - Known performance issues
- However, it can be used for early adopters
  - HLFIR (High Level FIR) is the default
  - No more `-flang-experimental-exec`
  - Can build SPEC® CPU 2017

# What is missing?

- Fortran 2003+ features
  - i.e. coarrays
- OpenMP 2.5+
- OpenACC
  - Not fully implemented yet

# Getting Involved

- Slack Workspace
  - <https://flang.lvm.org/docs/GettingInvolved.html#flang-slack-workspace>
- Calls
  - <https://flang.lvm.org/docs/GettingInvolved.html#calls>
- Discourse
  - <https://discourse.lvm.org/c/subprojects/flang/33>
- GitHub
  - <https://github.com/llvm/llvm-project>
- Docs
  - <https://flang.lvm.org/docs>



# Thank you

