linaro Connect 2025

LIS25-230 Quality Assurance of Flang Using the Fujitsu Compiler Test Suite

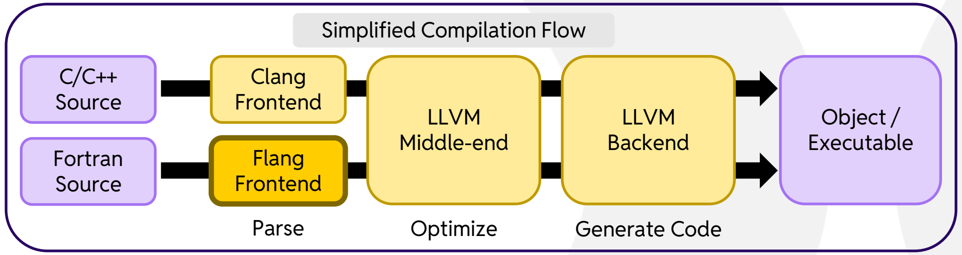
Takahiro Kawashima, Shun Kamatsuka
Fujitsu Limited

May 2025

What is Flang

Fortran frontend in the LLVM Project

- Open source
- Use LLVM Core as compiler middle-end and backend
- Clang for C/C++, Flang for Fortran
- Relatively "new" subproject
  - Approved by the LLVM Project in 2019
- Under active development
  - For Fortran/OpenMP language support and performance improvement
  - 2033 commits during 2024-04-01 – 2025-04-01 (in `flang` and `flang-rt` directories in the LLVM repo)
- Anticipated in HPC and AI domains



See MAD24-223 State of LLVM Flang Development [*1] and LLVM Fortran Levels Up: Goodbye flang-new, Hello flang! [*2] for details

*1 https://resources.linaro.org/en/resource/4SFvjFkHNfR28g2x3qgnUG
*2 https://blog.llvm.org/posts/2025-03-11-flang-new/

Existing Test Suite and CI in the LLVM Community

- “LLVM test-suite”
  - Large set of test programs mainly written in C, C++, and Fortran
    - Whole program test
    - Programs are compiled and executed, and the outputs are verified
  - Developed by the LLVM community
    - https://github.com/llvm/llvm-test-suite
- Pre-merge CI
  - Triggered by GitHub pull request creation
  - Light-weight correctness test using unit/regression tests in the source tree
- Post-merge CI
  - LLVM Project-official CIs
    - LLVM Buildbot for correctness test
    - LLVM Nightly Test for performance test
  - 3rd party CIs
    - Linaro Toolchain CI for correctness/performance tests
    - Some others
  - Run periodically for the LLVM development branch
  - The LLVM test-suite is used mainly for correctness test

Issues Surrounding the Quality of Flang

- The number of Fortran tests was insufficient
  - ~8500 Fortran tests in the LLVM test-suite
    - Most of them were imported from GFortran test suite
- Fortran tests were hardly run in the LLVM community
- Tests were hardly run with the fast-math compiler flag enabled, which allows optimizations that may introduce floating-point arithmetic precision errors
  - Important for Fortran; The fast-math flag is frequently used with Fortran applications
  - Difficult to verify the calculation result because of precision errors

Our Approach to Improve the Quality of Flang

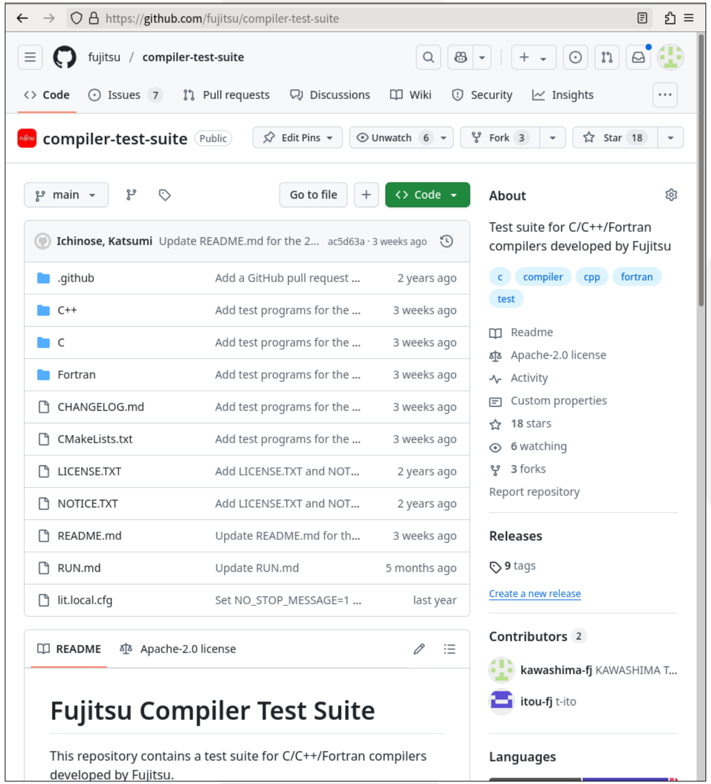
Assist Flang developers in developing high-quality Flang in an open environment

1. Prevent regressions
   - Release “Fujitsu Compiler Test Suite” to be used by Flang developers
2. Detect regressions
   - Run “Flang CI” using Fortran tests and with the fast-math flag
3. Fix existing bugs
   - Find, analyze, and report existing bugs

1. Fujitsu Compiler Test Suite (1/2)

linaro Connect

- Test suite for Fortran/C/C++ compilers
- Released on GitHub at the end of 2023 by Fujitsu
  - https://github.com/fujitsu/compiler-test-suite
- Check correctness (not performance)
- Accumulated over many years for testing proprietary Fujitsu compilers
  - ~64,000 for Fortran, ~26,000 for C, and ~5,000 for C++ (at version 2025-03)
  - Processor-dependent tests were ported to AArch64 from x86-64 and SPARC



https://github.com/fujitsu/compiler-test-suite

fujitsu / compiler-test-suite

Code · Issues 7 · Pull requests · Discussions · Wiki · Security · Insights

compiler-test-suite Public · Edit Pins · Unwatch 6 · Fork 3 · Star 18

main · Go to file · Code

Ichinose, Katsumi Update README.md for the 2... acd5d63a · 3 weeks ago

.github	Add a GitHub pull request ...	2 years ago
C++	Add test programs for the ...	3 weeks ago
C	Add test programs for the ...	3 weeks ago
Fortran	Add test programs for the ...	3 weeks ago
CHANGELOG.md	Add test programs for the ...	3 weeks ago
CMakeLists.txt	Add test programs for the ...	3 weeks ago
LICENSE.TXT	Add LICENSE.TXT and NOT...	2 years ago
NOTICE.TXT	Add LICENSE.TXT and NOT...	2 years ago
README.md	Update README.md for th...	3 weeks ago
RUN.md	Update RUN.md	5 months ago
lit.local.cfg	Set NO_STOP_MESSAGE=1 ...	last year

README · Apache-2.0 license

Fujitsu Compiler Test Suite

This repository contains a test suite for C/C++/Fortran compilers developed by Fujitsu.

About

Test suite for C/C++/Fortran compilers developed by Fujitsu

c · compiler · cpp · fortran · test

Readme
Apache-2.0 license
Activity
Custom properties
18 stars
6 watching
3 forks
Report repository

Releases

9 tags

Create a new release

Contributors 2

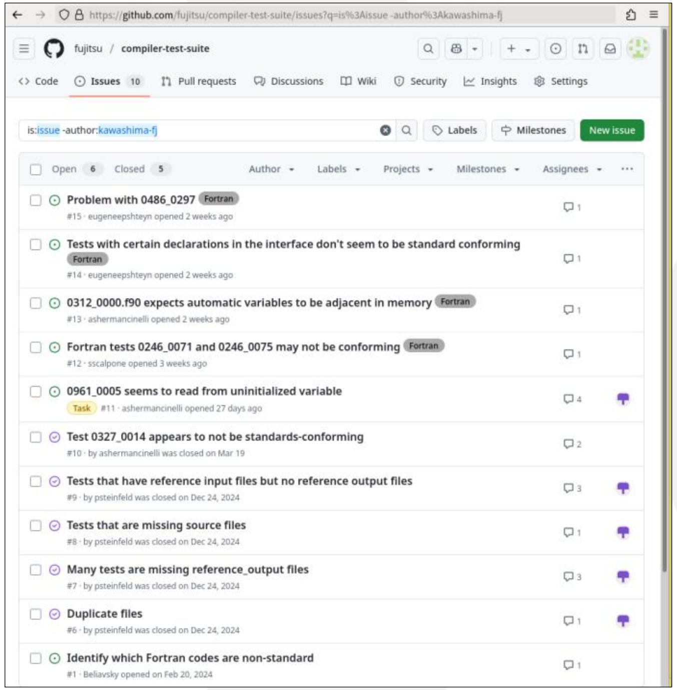
kawashima-fj KAWASHIMA T...
itou-fj t-ito

Languages

1. Fujitsu Compiler Test Suite (2/2)

- Various works were needed for the release
  - Adapt the structure to the LLVM test-suite's runner
  - Remove Fujitsu compiler-specific tests
  - Adjust implementation-defined behaviors to Flang
  - Eliminate internal information
    - Customer names, project names, bug IDs, …
  - Eliminate Japanese comments

- Used by Flang developers already
- Some feedback from the Flang community →


https://github.com/fujitsu/compiler-test-suite/issues?q=is%3Aissue -author%3Akawashima-fj

fujitsu / compiler-test-suite

is:issue -author:kawashima-fj

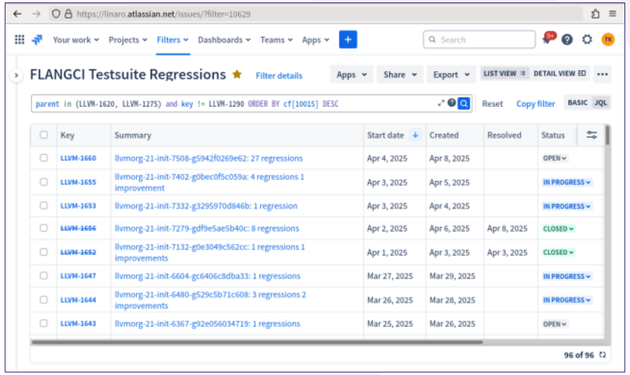
Open 6 Closed 5

- Problem with 0486_0297 Fortran — #15 · eugeneepshteyn opened 2 weeks ago — 1
- Tests with certain declarations in the interface don't seem to be standard conforming Fortran — #14 · eugeneepshteyn opened 2 weeks ago — 1
- 0312_0000.f90 expects automatic variables to be adjacent in memory Fortran — #13 · ashermancinelli opened 2 weeks ago — 1
- Fortran tests 0246_0071 and 0246_0075 may not be conforming Fortran — #12 · sscalpone opened 3 weeks ago — 1
- 0961_0005 seems to read from uninitialized variable — Task #11 · ashermancinelli opened 27 days ago — 4
- Test 0327_0014 appears to not be standards-conforming — #10 · by ashermancinelli was closed on Mar 19 — 2
- Tests that have reference input files but no reference output files — #9 · by psteinfeld was closed on Dec 24, 2024 — 3
- Tests that are missing source files — #8 · by psteinfeld was closed on Dec 24, 2024 — 1
- Many tests are missing reference_output files — #7 · by psteinfeld was closed on Dec 24, 2024 — 3
- Duplicate files — #6 · by psteinfeld was closed on Dec 24, 2024 — 1
- Identify which Fortran codes are non-standard — #1 · Beliavsky opened on Feb 20, 2024 — 1

Flang developers can run a large amount of Fortran tests by themselves

2. Flang CI (1/2)

- One of Linaro Toolchain CI
  - Utilize the existing platform of Linaro Toolchain CI
    - See LHR23-314-Toolchain Working Group CI and Regression Bisection *1 and MAD24-205 Overview of Linaro Toolchain CI configurations *2 for Linaro Toolchain CI
  - Collaboratively designed by Fujitsu and Linaro
- Post-merge CI to detect regressions introduced in the LLVM development branch
  - The cause commit is identified by automatic bisect
  - An author is notified of the detected regression by email
- Run on AArch64 machines (AWS Gravition3)
- Operation started on July 2024



FLANGCI Testsuite Regressions

```
parent in (LLVM-1620, LLVM-1275) and key != LLVM-1290 ORDER BY cf[10015] DESC
```

Key	Summary	Start date	Created	Resolved	Status
LLVM-1660	llvmorg-21-init-7508-g5942f0269e62: 27 regressions	Apr 4, 2025	Apr 8, 2025		OPEN
LLVM-1655	llvmorg-21-init-7402-g0bec0f5c059a: 4 regressions 1 improvement	Apr 3, 2025	Apr 5, 2025		IN PROGRESS
LLVM-1653	llvmorg-21-init-7332-g3295970d846b: 1 regression	Apr 3, 2025	Apr 4, 2025		IN PROGRESS
~~LLVM-1656~~	llvmorg-21-init-7279-gdf9e5ae5b40c: 8 regressions	Apr 2, 2025	Apr 6, 2025	Apr 8, 2025	CLOSED
~~LLVM-1652~~	llvmorg-21-init-7132-g0e3049c562cc: 1 regressions 1 improvements	Apr 1, 2025	Apr 3, 2025	Apr 3, 2025	CLOSED
LLVM-1647	llvmorg-21-init-6604-gc6406c8dba33: 1 regressions	Mar 27, 2025	Mar 29, 2025		IN PROGRESS
LLVM-1644	llvmorg-21-init-6480-g529c5b71c608: 3 regressions 2 improvements	Mar 26, 2025	Mar 28, 2025		IN PROGRESS
LLVM-1643	llvmorg-21-init-6367-g92e056034719: 1 regressions	Mar 25, 2025	Mar 26, 2025		OPEN

96 of 96

*1 https://resources.linaro.org/en/resource/jMMmptWaoCaVghsLDXvP3P
*2 https://resources.linaro.org/en/resource/ooVkEuWFFnBXw6TNw2SLJZ

2. Flang CI (2/2)

- Use the Fujitsu Compiler Test Suite
  - Also run C/C++ tests, in addition to Fortran tests
- Four configurations
  1. Optimization with fast-math, SVE VLA
  2. Optimization with fast-math, SVE VLS, LTO
  3. Optimization without the fast-math flag
     - If FAIL with 1 and 2 but PASS with 3: Likely a bug related to fast-math or a false detection due to precision errors
  4. No optimization
     - If FAIL with 1, 2, and 3 but PASS with 4: Likely a bug in optimizations

Flang developers become aware of regressions

3. Bug Reporting

1. Run the Fujitsu Compiler Test Suite
   - Including Fujitsu-proprietary tests not-yet-included in the public test suite
   - Only Fortran tests
2. Analyze the detected failures
   - Confirm the validity of the tests by referring to the Fortran standards
   - Compare Flang with GFortran and Intel Fortran Compiler
   - Create minimal reproducers
3. Report bugs as GitHub issues
4. Fix the reported bugs
   - By Linaro engineers and other Flang developers

Fujitsu engineers

LLVM community

Flang developers become aware of existing bugs

Results of Our Activities

Flang CI

- 49 regressions were detected
- All of them were fixed

Bug reporting

- 269 existing bugs were reported
- 232 of them (86%) were resolved

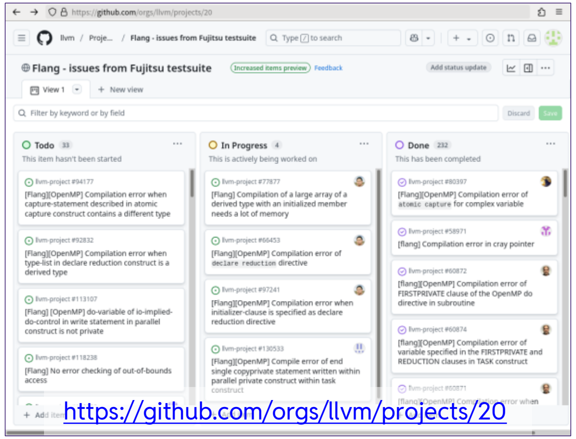
Test result

- The failure rate using the Fujitsu Compiler Test Suite *1 is less than 1%

Failures detected by Flang CI
(as of 2025-04-28)

Cause		Open	Closed	Total
Bug	New regression	0	49	49
	Latent bug	1	3	4
Not Bug	Revert commit	0	3	3
	Problem of test	7	14	21
	Precision error	10	1	11
Other		0	5	5
Uninvestigated		7	0	7
Total		25	75	100

Bug reports from Fujitsu
(as of 2025-04-28)


https://github.com/orgs/llvm/projects/20
llvm / Proje... / Flang - issues from Fujitsu testsuite
Type / to search
Increased items preview
Feedback
Add status update
View 1
New view
Filter by keyword or by field
Discard
Save
Add item

Flang - issues from Fujitsu testsuite

Todo 33 — This item hasn't been started

- llvm-project #94177 — [Flang][OpenMP] Compilation error when capture-statement described in atomic capture construct contains a different type
- llvm-project #92832 — [Flang][OpenMP] Compilation error when type-list in declare reduction construct is a derived type
- llvm-project #113107 — [Flang] [OpenMP] do-variable of io-implied-do-control in write statement in parallel construct is not private
- llvm-project #118238 — [Flang] No error checking of out-of-bounds access

In Progress 4 — This is actively being worked on

- llvm-project #77877 — [Flang] Compilation of a large array of a derived type with an initialized member needs a lot of memory
- llvm-project #66453 — [Flang][OpenMP] Compilation error of `declare reduction` directive
- llvm-project #97241 — [Flang][OpenMP] Compilation error when initializer-clause is specified as declare reduction directive
- llvm-project #130533 — [Flang][OpenMP] Compile error of end single copyprivate statement written within parallel private construct within task construct

Done 232 — This has been completed

- llvm-project #80397 — [Flang][OpenMP] Compilation error of `atomic capture` for complex variable
- llvm-project #58971 — [flang] Compilation error in cray pointer
- llvm-project #60872 — [flang][OpenMP] Compilation error of FIRSTPRIVATE clause of the OpenMP do directive in subroutine
- llvm-project #60874 — [flang][OpenMP] Compilation error of variable specified in the FIRSTPRIVATE and REDUCTION clauses in TASK construct
- llvm-project #60871 — [flang][OpenMP] Compilation error when

https://github.com/orgs/llvm/projects/20

Flang test result
(as of 2025-03-10)

Standards	Pass	NYI *2	Fail	Fail %
77-95	42202	1198	73	0.17
2003	22546	2729	58	0.23
2008	12523	660	81	0.61
2018	359	51	2	0.49

*1 Including Fujitsu-proprietary tests not-yet-included in the public test suite

*2 Not Yet Implemented: Compile-time message is emitted to indicate the feature is not yet implemented (not a bug)

Future Plan

- Improve the reliability of the test suite and the CI
  - Fix issues in the Fujitsu Compiler Test Suite (non-conforming, implementation-defined, ...)
    - Now working. Gradually resolving.
  - Reduce false detections by precision errors of floating-point arithmetic in Flang CI
    - Now working. Hopefully completed in this year (2025).
- Enhance the test suite
  - Increase tests by making Fujitsu internal tests runnable with the LLVM test-suite's runner
    - Now working. Gradually increasing. Hopefully completed in this year (2025).
- Promote wider adoption
  - Integrate the Fujitsu Compiler Test Suite into the LLVM test-suite
    - After improving the test suite reliability. Need consensus with the community.
- Test for newer Standards (Fortran 2018, 2023)
    - Discuss with the community.

Summary

- Flang is under active development

- Fujitsu is collaborating with Linaro to improve the quality of Flang
    1. Prevent regressions: “Fujitsu Compiler Test Suite”
    2. Detect regressions: “Flang CI”
    3. Fix existing bugs: Find, analyze, and report existing bugs

- Many bugs have been detected and fixed already

Acknowledge

This presentation is based on results obtained from a project, JPNP21029, subsidized by the New Energy and Industrial Technology Development Organization (NEDO).

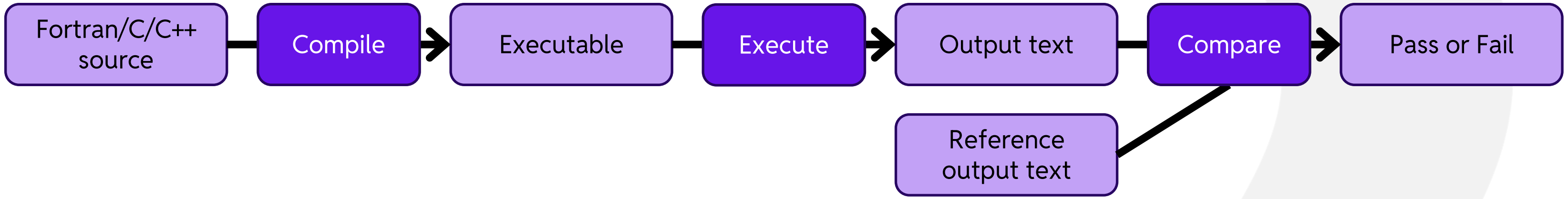
linaro Connect 2025

Thank You!

Backup Slides

Characteristics of the Fujitsu Compiler Test Suite (1/4)

- Whole program test
  - Programs are compiled and executed, and the outputs are verified
    - Like “SingleSource” tests in the LLVM test-suite
  - Compile-only tests also exist



- Single-source tests only (currently)
  - Multiple-source tests will be added in the future
  - Typically, 10–100 lines

Characteristics of the Fujitsu Compiler Test Suite (2/4)

- Tests for Fortran, C, and C++, with OpenMP

Language	Fortran	C	C++
# of tests	~64,000	~26,000	~5,000
% of OpenMP tests	7%	6%	6%

Fortran Standards	77/95	2003	2008	2018
% of tests	51%	32%	16%	≤ 1%

(at version 2025-03)

  - No tests for OpenACC, CUDA, or other external extensions
  - (FYI: the LLVM test-suite contains ~8,500 Fortran tests, most of which were imported from the GFortran test suite)
- Not categorized in directories
- Many code patterns used in the HPC domain
  - Floating-point arithmetic, loops, ...

Characteristics of the Fujitsu Compiler Test Suite (3/4)

- Fortran tests
  - No tests for Fortran coarray
    - Fujitsu has internal tests, which can be released if the community has interests
  - All features of Fortran 2008 and earlier
    - Only basic tests for Fortran 2008; Insufficient combination tests
  - Some features of Fortran 2018
    - Assumed type, assumed rank, etc.
  - Include extensions that Flang and the Fujitsu Fortran Compiler have in common

Characteristics of the Fujitsu Compiler Test Suite (4/4)

- Check correctness (not performance)
  - Language standard conformance
  - Optimization correctness
  - Code generation correctness
  - Fortran and OpenMP runtime libraries' behavioral correctness
- No dependency on external libraries (MPI or math libraries)
- Include Linux/AArch64-specific tests
  - Disabling Linux/AArch64-specific tests on other platforms is not yet implemented
  - Most tests will be run on machines other than Linux/AArch64
  - No architecture-specific tests other than AArch64
  - Only tested on Linux/AArch64

Where the Fujitsu Compiler Test Suite can be used

- You can use the Fujitsu Compiler Test Suite as a supplement to the LLVM test-suite, especially for:
    - Flang frontend/runtime
    - OpenMP runtime
    - Loop optimization

How to Use the Fujitsu Compiler Test Suite

- Use it as an add-on to the LLVM test-suite
  - If you can run “SingleSource” tests in the LLVM test-suite, it’s easy
  - Details in https://github.com/fujitsu/compiler-test-suite/blob/main/RUN.md

Command example

```
git clone https://github.com/llvm/llvm-test-suite.git
git clone https://github.com/fujitsu/compiler-test-suite.git
    llvm-test-suite/Fujitsu
mkdir llvm-test-suite-build
cd llvm-test-suite-build

cmake -G Ninja
    -D CMAKE_C_COMPILER=<path to llvm build>/bin/clang
    -D CMAKE_Fortran_COMPILER=<path to llvm build>/bin/flang
    -D TEST_SUITE_SUBDIRS=Fujitsu
    -D TEST_SUITE_FORTRAN=ON
    -C ../llvm-test-suite/cmake/caches/O0.cmake
    ../llvm-test-suite
ninja -k 0
lit -o results.json .
```

(Magenta/bold lines are required)

Directory structure

```
llvm-test-suite
  |- SingleSource
  |- MultiSource
  |- Fortran
  |- CMakeLists.txt
  |- ...
  |- Fujitsu    ← Added
      |- Fortran
      |- C
      |- C++
      |- CMakeLists.txt
      |- lit.local.cfg
      |- ...
```