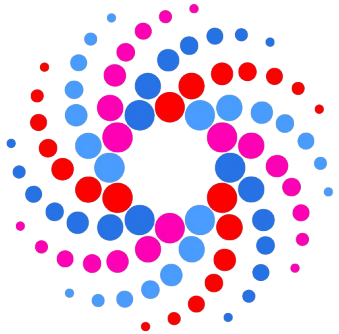
OpenMP®

SC'22 Booth Talk Series



SC22

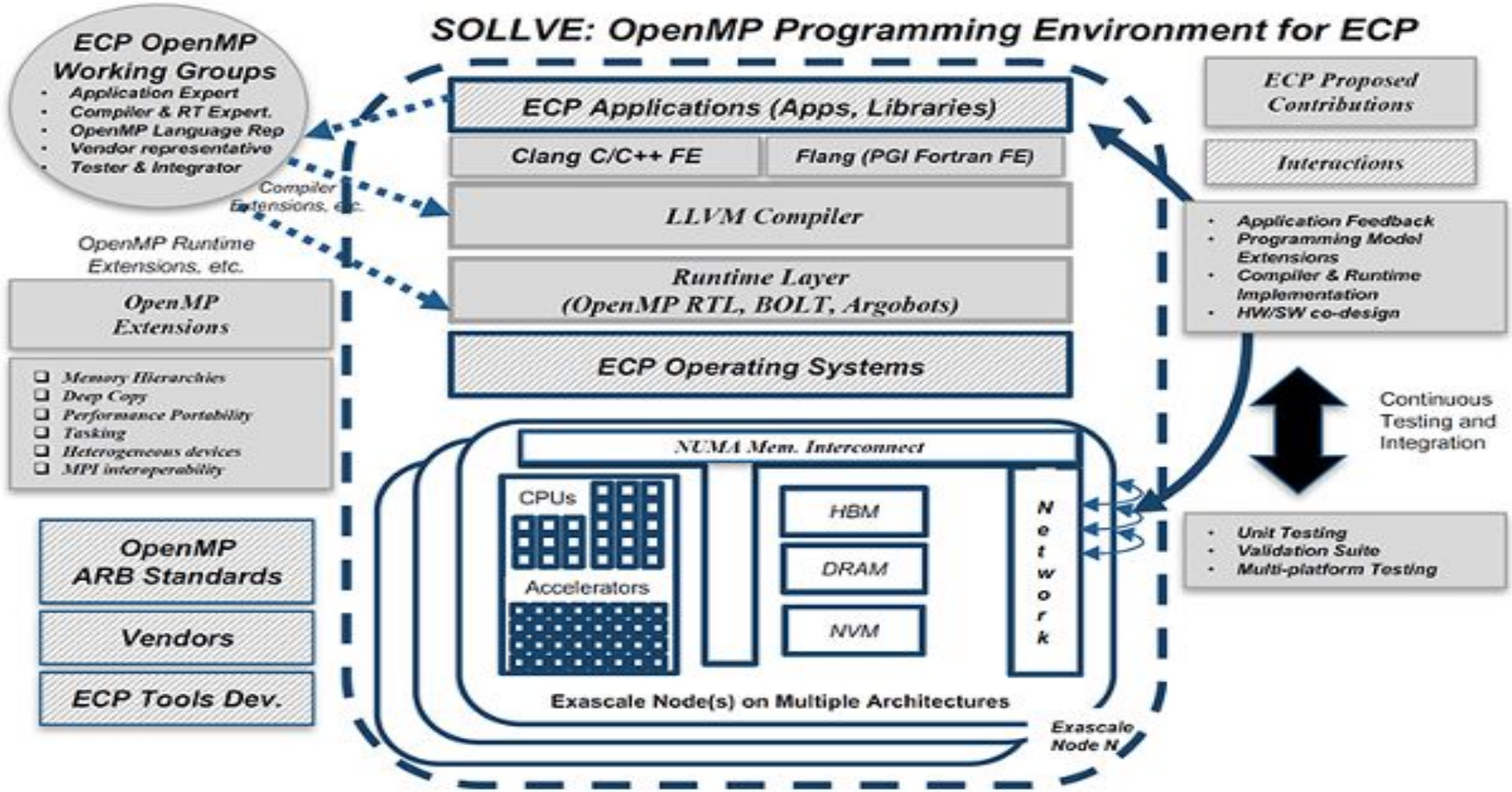
Dallas, TX | hpc accelerates.

SOLLVE OpenMP Validation and Verification Testsuite

ECP Project WBS 2.3.1.13 STPM15

Swaroop Pophale (ORNL)
Thomas Huber (University of Delaware)

Exascale Computing Project: SOLLVE



Team Members

- Swaroop Pophale
- Seyong Lee
- David E. Bernholdt

- Thomas Huber
- Nolan Baker
- Nikhil Rao
- Kristina Holsapple
- Michael Carr
- Jaydon Reap
- Felipe Cabarcas
- Sunita Chandrasekaran (PI)

OpenMP Validation and Verification Testsuite

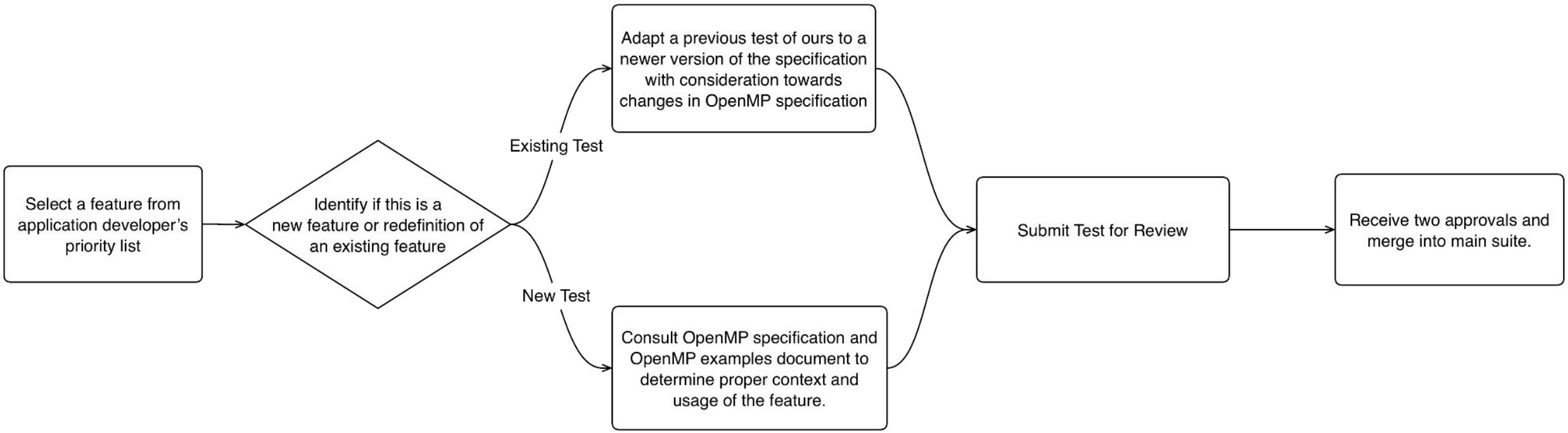
- What is it?
  - Suite of tests encompassing of the OpenMP features (directives, constructs, runtime calls)

- Why is it?
  - Evaluate OpenMP readiness of various systems and their compilers
  - Help Compiler Developers
    - Peer reviewed, publicly available, up-to-date
    - Test conformance for different releases
  - Help Application Developers
    - Before attempting to utilize a new feature, take a look

Latest OpenMP Specification: 5.2

- Released November 2021
- Around 27 modifications (new features, deprecated features, behavior changes)
  - Fortran specific features
    - Metadirectives, assumption directives, nothing directive, error directives, and loop transformation constructs for pure procedures
    - allocators construct
    - dispatch construct with end directive
- Few-to-none of these modifications to the specification are implemented in popular compilers
  - GCC provides status here https://gcc.gnu.org/onlinedocs/libgomp/OpenMP-5_002e2.html#OpenMP-5_002e2
  - LLVM provides status here https://clang.llvm.org/docs/OpenMPSupport.html
- How do we generate tests from scratch, especially when there are no implementations

Writing a test from scratch (Our abridged workflow)



Running the test suite

1. Clone our repo https://github.com/SOLLVE/sollve_vv
2. Setup your environment (install or ‘module load’ your compilers)
3. Use our make commands to compile and run a single test or the whole suite

Running a single test:

```
make CC=gcc CXX=g++ FC=gfortran VERBOSE=1 VERBOSE_TESTS=1 LOG=1 LOG_ALL=1 SYSTEM=summit
DEVICE_TYPE=nvidia OMP_VERSION=5.1 SOURCES=name_of_test all
```

Running the entire suite:

```
make CC=gcc CXX=g++ FC=gfortran VERBOSE=1 VERBOSE_TESTS=1 LOG=1 LOG_ALL=1 SYSTEM=summit
DEVICE_TYPE=nvidia OMP_VERSION=5.1 SOURCES=* all
```

Compilers We Test

- AMD
- GNU
- HPE/Cray
- IBM
- LLVM
- NVIDIA

Systems of Interest

- Summit (Oak Ridge National Lab)
- Perlmutter (NERSC)
- Spock (Oak Ridge National Lab)

Results Reporting

- Our testsuite can easily be included in your weekly/nightly runs
  - As simple as running ‘make report_json’
  - Run results nicely formatted in .json
  - Easily consumable by your visualization tools (spreadsheets, Elasticsearch)
  - Can also run ‘make report_summary’ to get quick overview of your total pass/fail numbers

Coverage

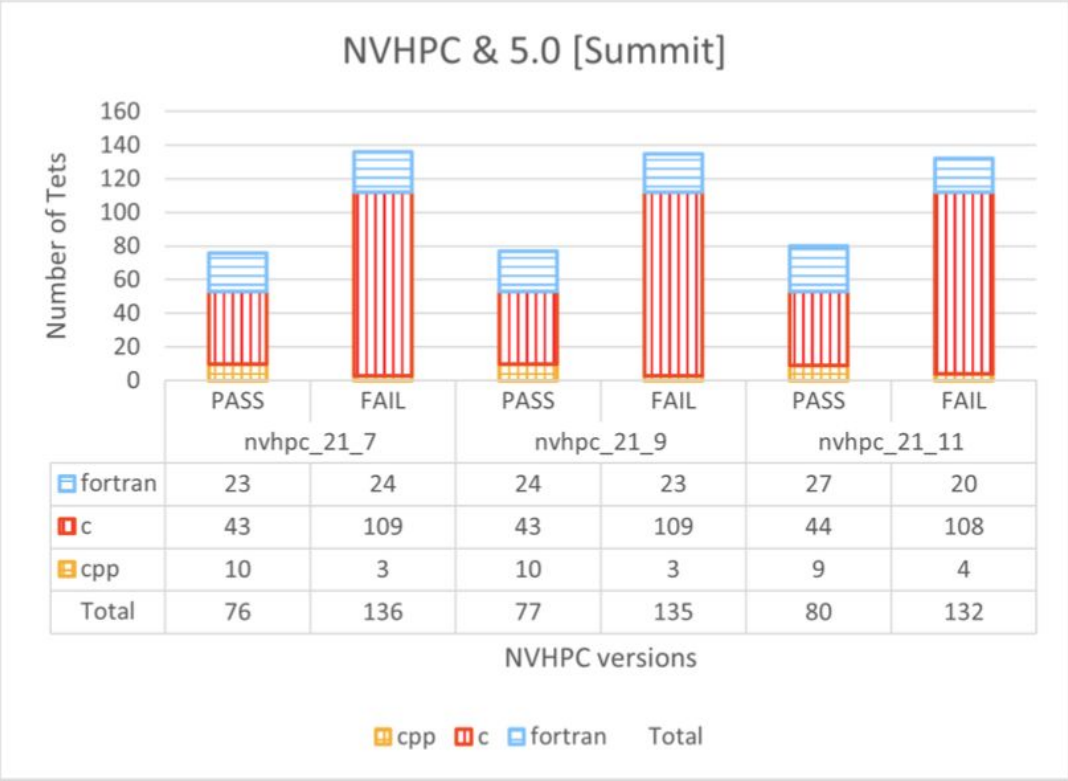
OpenMP 5.0
- 258 tests
  - 100% coverage for C/C++
  - 70% coverage for Fortran

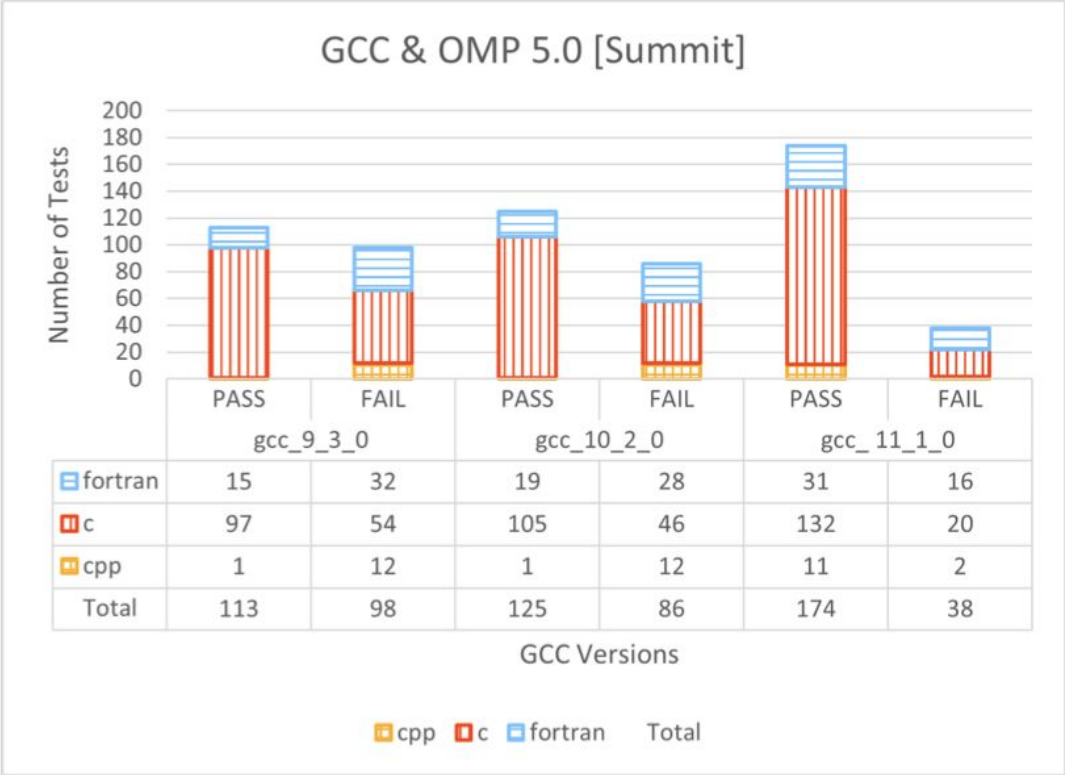
OpenMP 5.1
- 45 tests
  - 85% coverage for C/C++
  - 5% coverage for Fortran

OpenMP 5.2
- 6 tests
  - 20% overall coverage

Test creation priority is based on needs of application developers.

Progression of Compilers Over Time





Community Interaction

- Recent Success Stories
  - Test had a ‘slip-up’ that led to a patch in GCC https://gcc.gnu.org/PR102972
  - Test led to issue filed for inclusion in OpenMP 6.0 specification
  - Test led to a new description of the threadprivate directive.
- OpenMP Official Examples Document has adapted one of our test cases

Ways to collaborate

Get involved in our project through Github!

- https://github.com/SOLLVE/sollve_vv

Acknowledgment

Work supported by the U.S. Department of Energy, Office of Science, the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration under contract number DE-AC05-00OR22725.