OpenMP®

SC'20 Booth Talk Series

SOLLVE OpenMP Validation and Verification Effort

ECP Project WBS 2.3.1.13 STPM15

Swaroop Pophale (ORNL)

ECP EXASCALE COMPUTING PROJECT

SOLLVE OpenMP V&V: The Team

ECP Project WBS 2.3.1.13 STPM15

Swaroop Pophale

David E. Bernholdt

Josh Davis

Thomas Huber

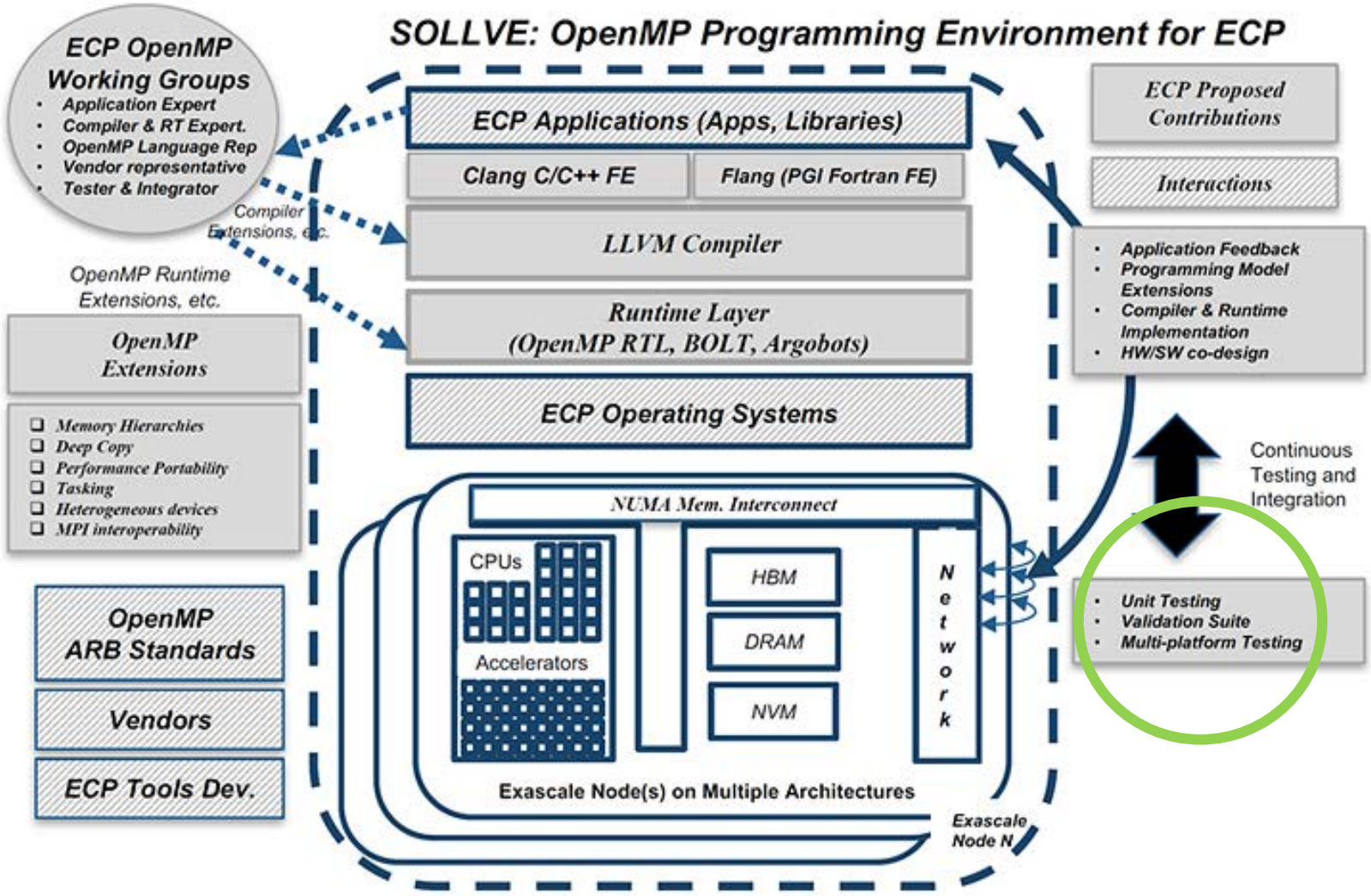
Sunita Chandrasekaran

Outline

- Introduction
  - What is SOLLVE ?
  - Scope and Intent of the Validation and Verification Suite
  - OpenMP Offload (4.x+) Machine Model
  - OpenMP 5.0 New Features
- Validation and Verification Suite
  - Test design process
  - Examples
  - Infrastructure design
- Sample Results
- Success Stories
- Ways to collaborate

3

Introduction: The SOLLVE ECP Project



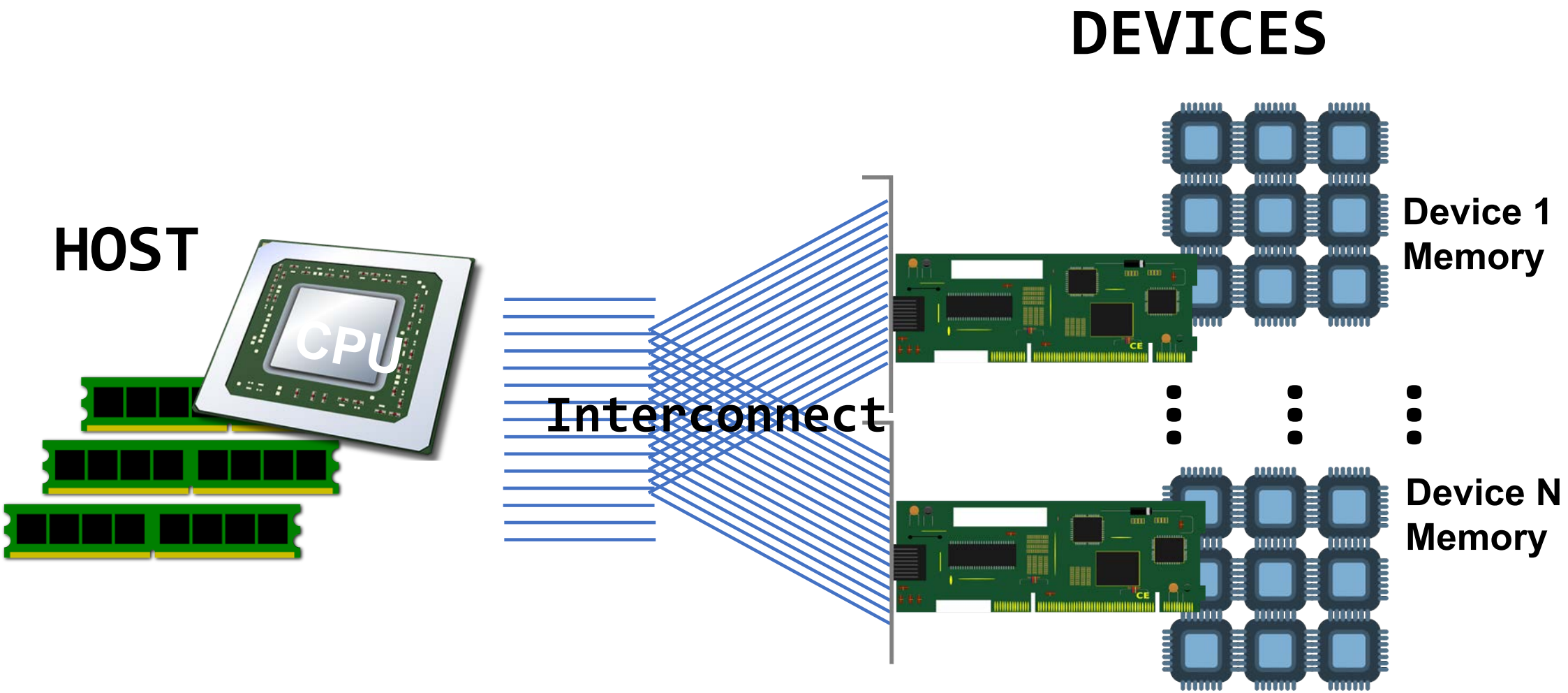
ECP EXASCALE COMPUTING PROJECT

Introduction: Scope and Intent of V&V suite

- Verify the status of OpenMP implementations across ECP platforms
- Evaluation of OpenMP functionality and performance on target architectures
- Check implementations' conformance to the OpenMP standard
- Tests and kernels primarily motivated by ECP Applications
- Verify platforms are ready for OpenMP applications

EXASCALE COMPUTING PROJECT

Introduction: OpenMP Offload (4.x+) Machine Model

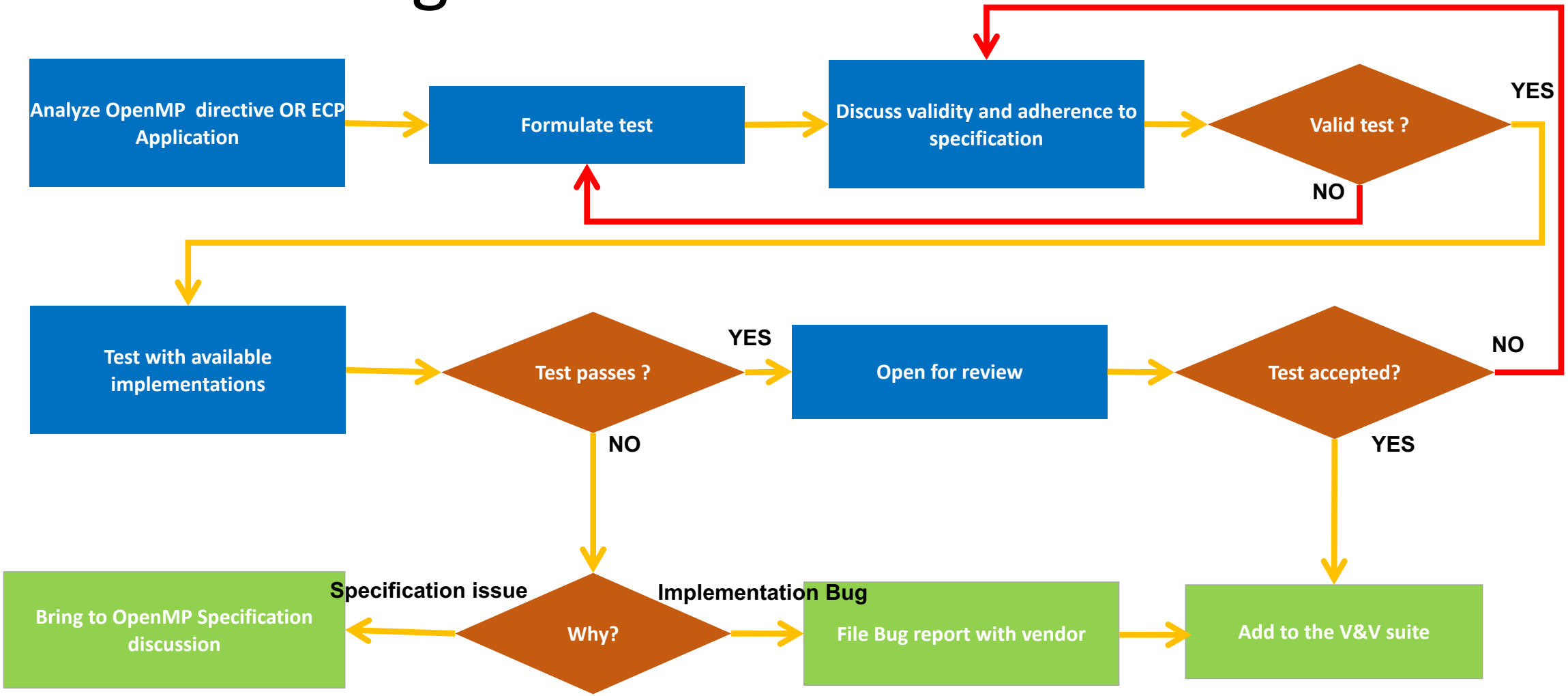


Introduction: OpenMP 5.0 New Features

- OpenMP contexts, metadirective, and declare variant
- Addition of requires directive, including support for unified shared memory
- Memory allocators and support for deep memory hierarchies
- Descriptive loop construct
- Release/acquire semantics added to memory model
- First (OMPT) and third (OMPD) party tool support
- Completed support for Fortran 2003
- Added support for Fortran 2008, C11, C++11, C++14 and C++17

ECP EXASCALE COMPUTING PROJECT

V&V: Test Design Process



ECP EXASCALE COMPUTING PROJECT

V&V Example 1: Understanding Spec.

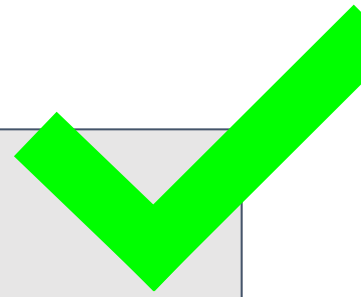
```
#pragma omp target data map(tofrom: a[0:ARRAY_SIZE]) {
#pragma omp target teams distribute
    ...
}
```

```
#pragma omp target data map(tofrom: a[0:ARRAY_SIZE/2]) {
#pragma omp target teams distribute
    ...
}
```

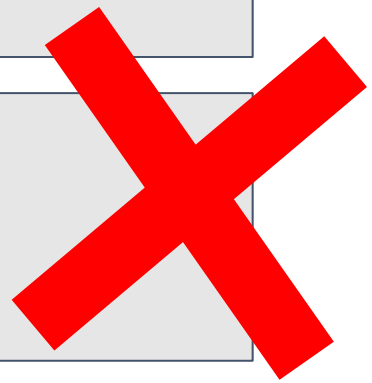

ECP EXASCALE COMPUTING PROJECT

V&V Example 1: Understanding Spec.

```
#pragma omp target data map(tofrom: a[0:ARRAY_SIZE]) {
#pragma omp target teams distribute
    ...
}
```



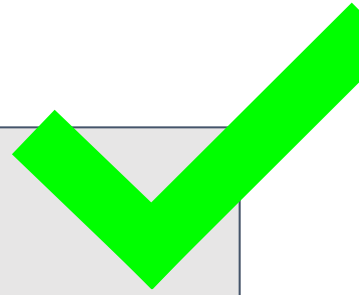
```
#pragma omp target data map(tofrom: a[0:ARRAY_SIZE/2]) {
#pragma omp target teams distribute
    ...
}
```



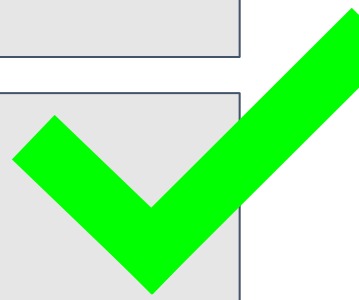
10

V&V Example 1: Understanding Spec.

```
#pragma omp target data map(tofrom: a[0:ARRAY_SIZE]) {
#pragma omp target teams distribute map(alloc: a[0:ARRAY_SIZE])
    ...
}
```



```
#pragma omp target data map(tofrom: a[0:ARRAY_SIZE/2]) {
#pragma omp target teams distribute map(alloc: a[0:ARRAY_SIZE/2])
    ...
}
```



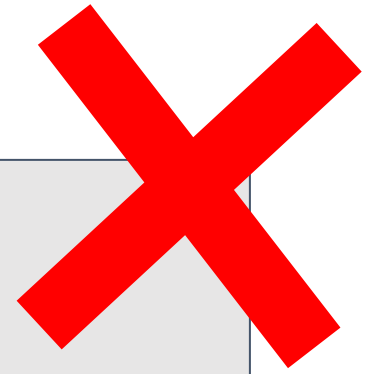
- Specification doesn't explicitly mention this requirement, but it is an implication arising from multiple interacting sections of the specification
- Prone to misinterpretation by vendors — XLC compiles and runs with no complaint, Clang gives a compile error explaining the problem, and GCC gives a runtime error

V&V Example 2: Ambiguous Specification

Restrictions

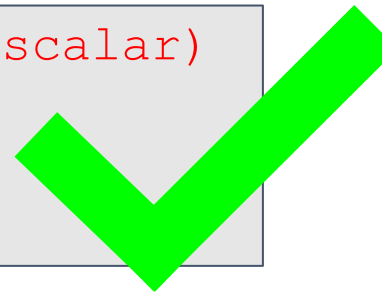
- A list item cannot appear in both a **map** clause and a data-sharing attribute clause on the same construct.

(From map clause specification, pg. 218)



```
#pragma omp target teams distribute shared(share) map(tofrom:share)
for (int x = 0; x < SIZE; ++x) {
        ...
}
```

```
#pragma omp target teams distribute shared(share) defaultmap(tofrom:scalar)
for (int x = 0; x < SIZE; ++x) {
        ...
}
```



12

V&V Example 3: Test from Application

```
#pragma omp target map(tofrom: a, sum) depend(out: a) nowait
  {
    for (i = 0; i < N; i++) {
       sum++;
    }
    a += 1;
  }

#pragma omp task depend(in: a) shared(a,errors)
  {
     if(a != 1) {
       errors += 1;
     }
  }
  #pragma omp taskwait
  if (sum != N) {
    errors++;
  }
```

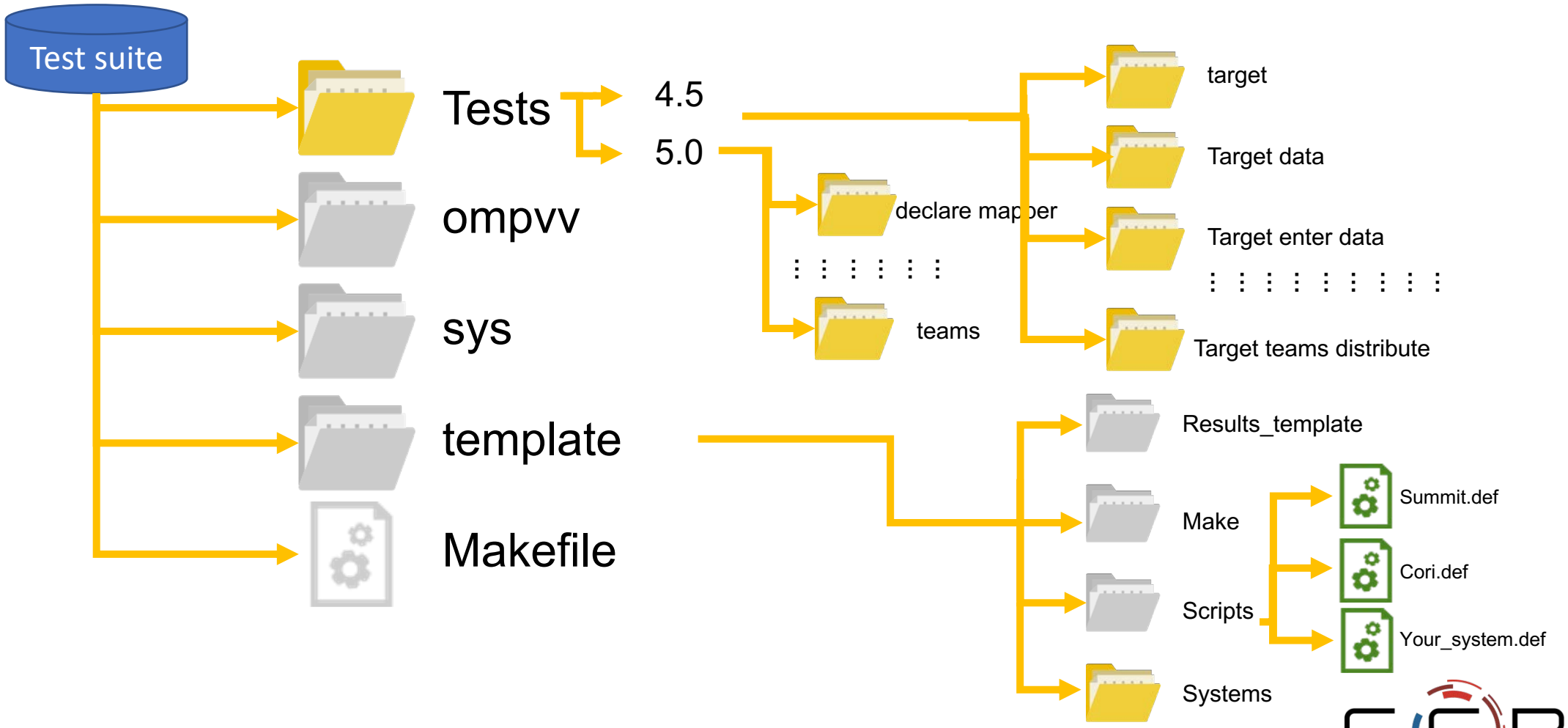

13

V&V: Infrastructure Design

- Our infrastructure is based on a **Makefile + scripts**
- Design parameters:
  - Portability across multiple **compilers** and **systems** and easy to use
    - Support for different compiler options
    - Support for Lua-like Modules and batch schedulers
  - Fast test addition and modification
  - Divided compilation and execution phases
  - Subset of tests selection for partial execution

EXASCALE COMPUTING PROJECT

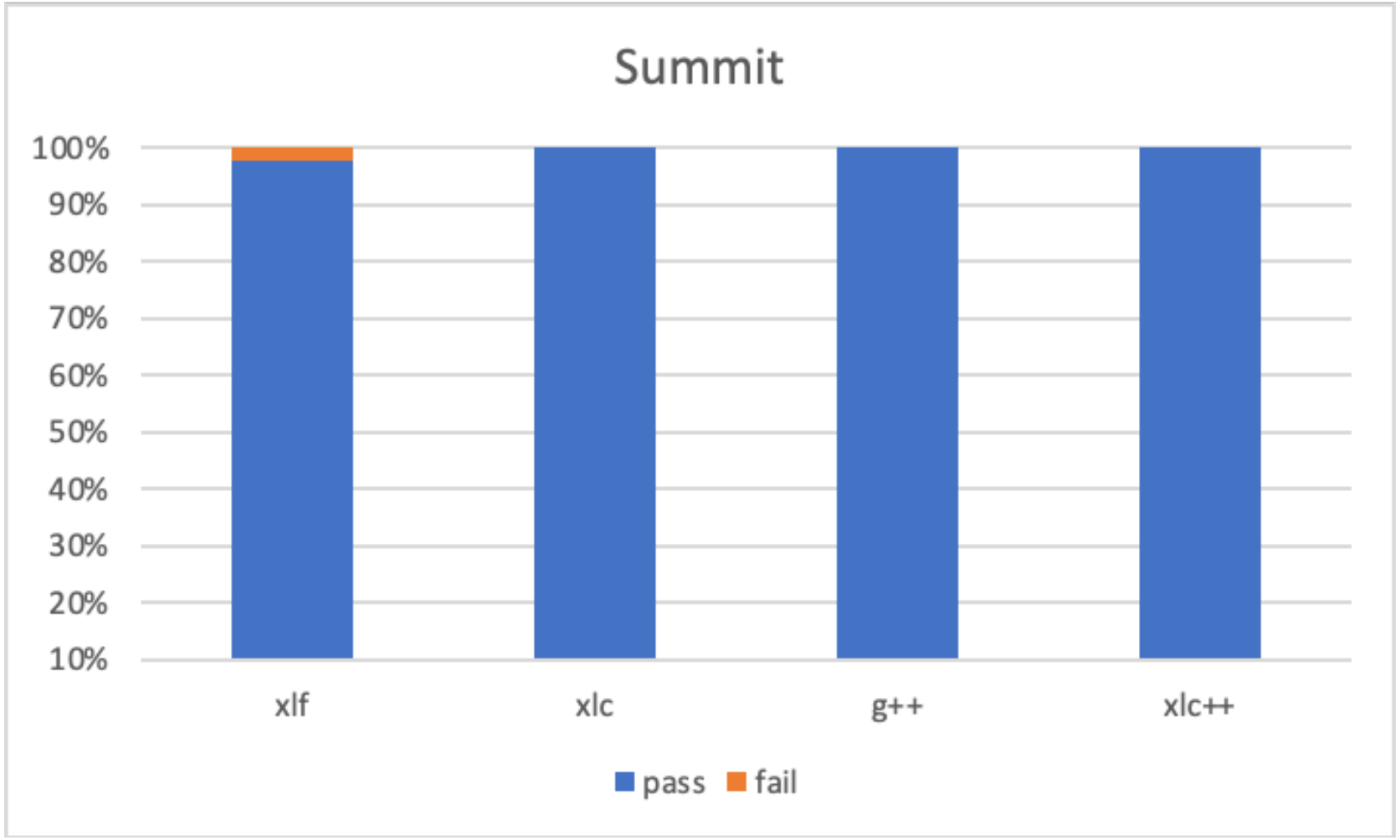
V&V: Infrastructure Design (cont.)



15

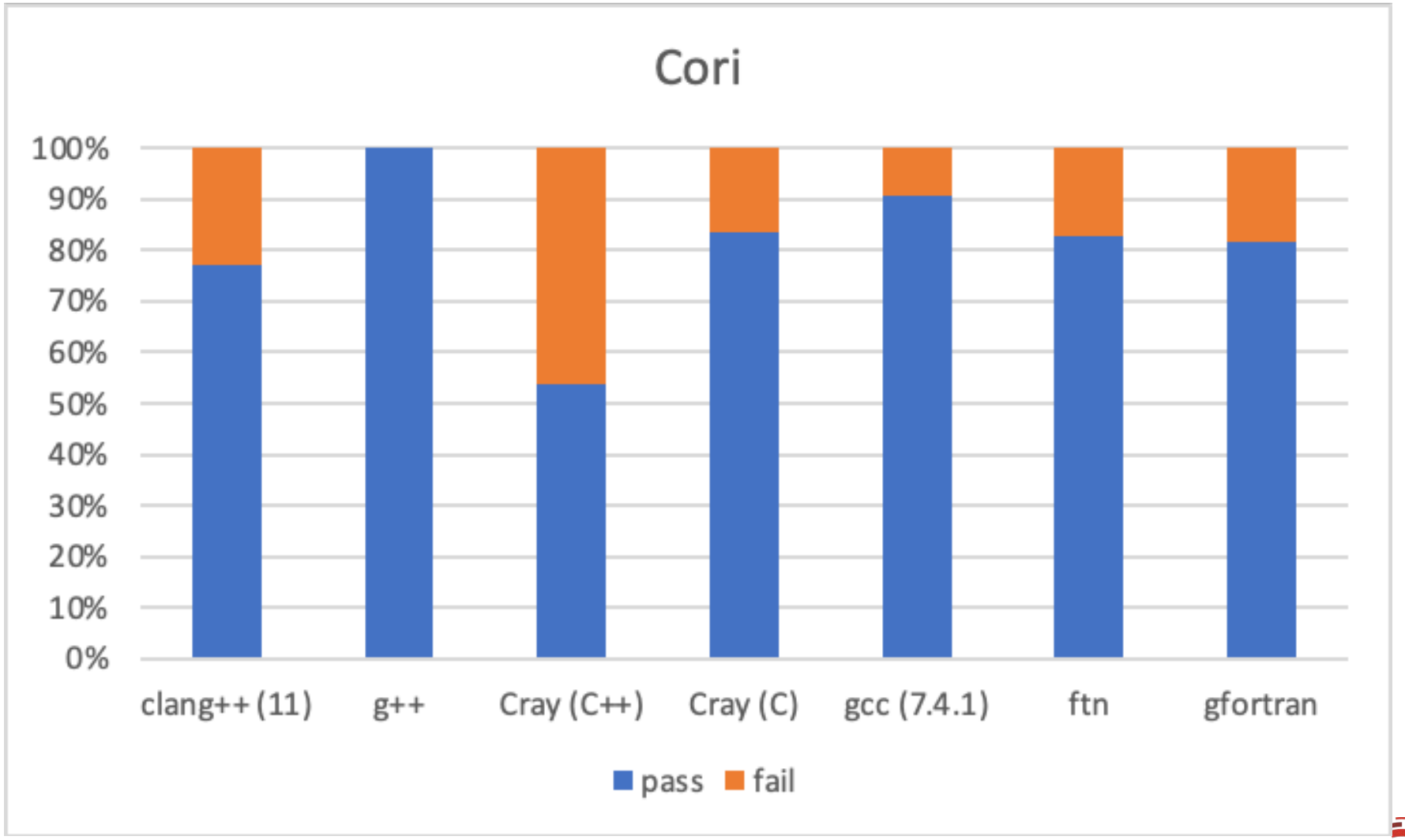
EXASCALE COMPUTING PROJECT

Sample Results: Summit



EXASCALE COMPUTING PROJECT

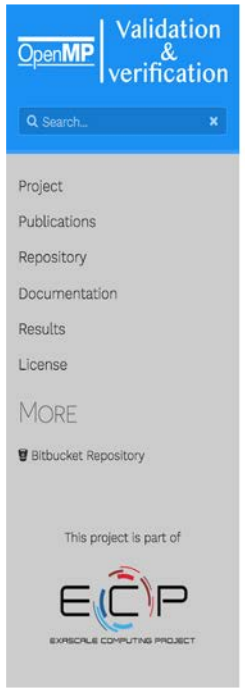
Sample Results: Cori

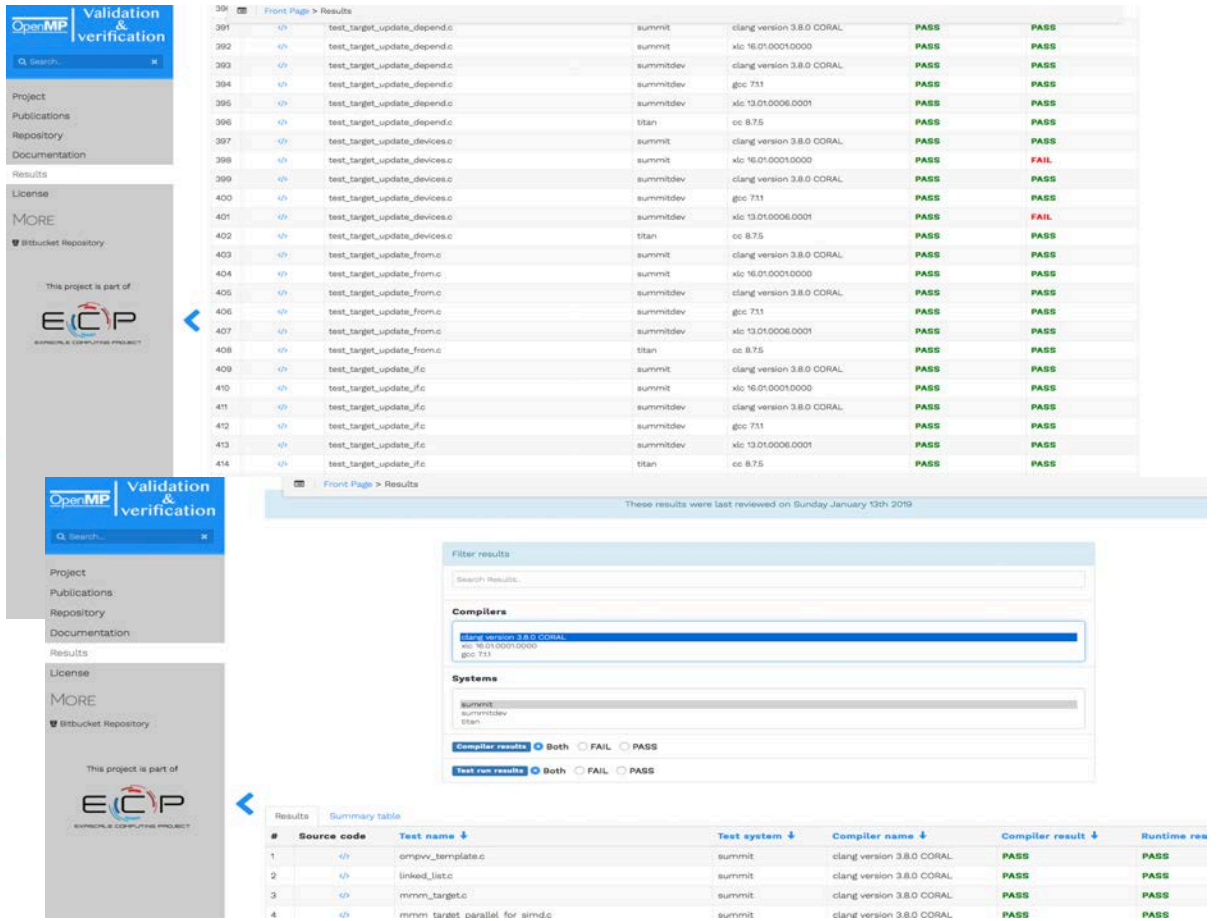


EXASCALE COMPUTING PROJECT

Sample Results: Current status of OMP Compilers

https://crpl.cis.udel.edu/ompvvsollve/





EXASCALE COMPUTING PROJECT

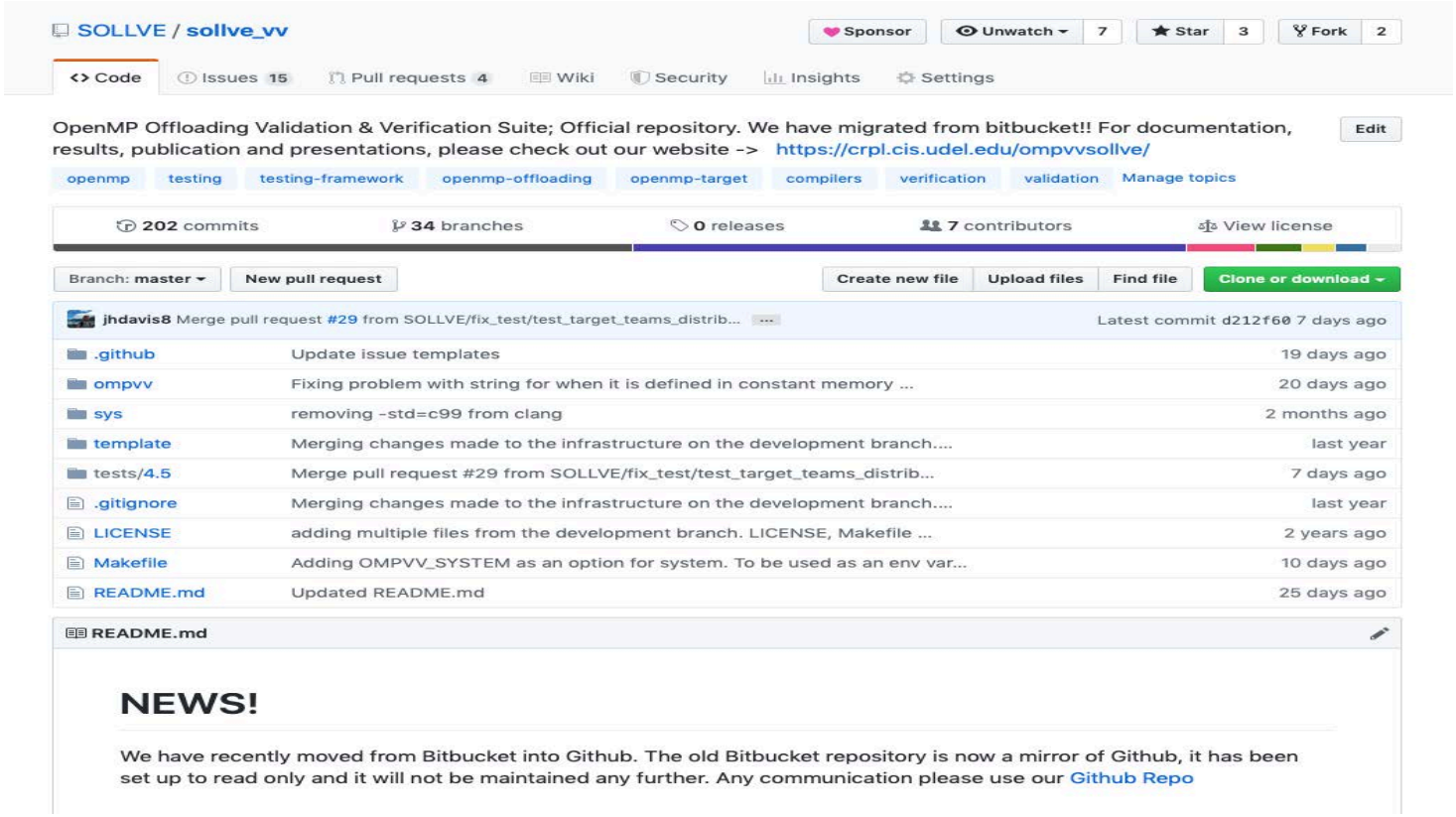
Success Stories

- Cray and AMD are actively using SOLLVE V&V tests to verify correctness and coverage of new features introduced in OpenMP 4.5 and 5.0 (on-going effort).
- Vendors are actively engaging to improve coverage and bug fixes.
- Implementation bugs were identified in the GCC, IBM, and Cray implementations and have been brought to the vendor's attention.
- OLCF is using SOLLVE V&V as part of acceptance tests for exascale ECP platform Frontier.
- NERSC has integrated SOLLVE V&V for testing OpenMP 4.5/5.0 implementations.

19

Ways to Collaborate: GitHub

https://github.com/SOLLVE/sollve_vv



ECP EXASCALE COMPUTING PROJECT

Ways to Collaborate: GitHub

- Need to check the behavior of YOUR OpenMP application ?
  - Create an application kernel
  - SUBMIT an Issue OR CREATE a PR on **https://github.com/SOLLVE/sollve_vv**

- Found something we missed ?
  - SUBMIT an Issue OR CREATE a PR on **https://github.com/SOLLVE/sollve_vv**

21

Ways to Collaborate: Contact information

Swaroop Pophale (pophaless@ornl.gov)
David E. Bernholdt (bernholdtde@ornl.gov)

Josh Davis (jhdavis@udel.edu)
Thomas Huber (thuber@udel.edu)
Sunita Chandrasekaran (schandra@udel.edu)

Acknowledgement

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.**

OpenMP®

SC'20 Booth Talk Series

For the OpenMP specification, tutorials, forum, reference guides, and links to other resources, visit **www.openmp.org**

ECP EXASCALE COMPUTING PROJECT