TestCov:

Robust Test-Suite Execution and Coverage Measurement

Thomas Lemberger

Joint work with Dirk Beyer

LMU Munich, Germany

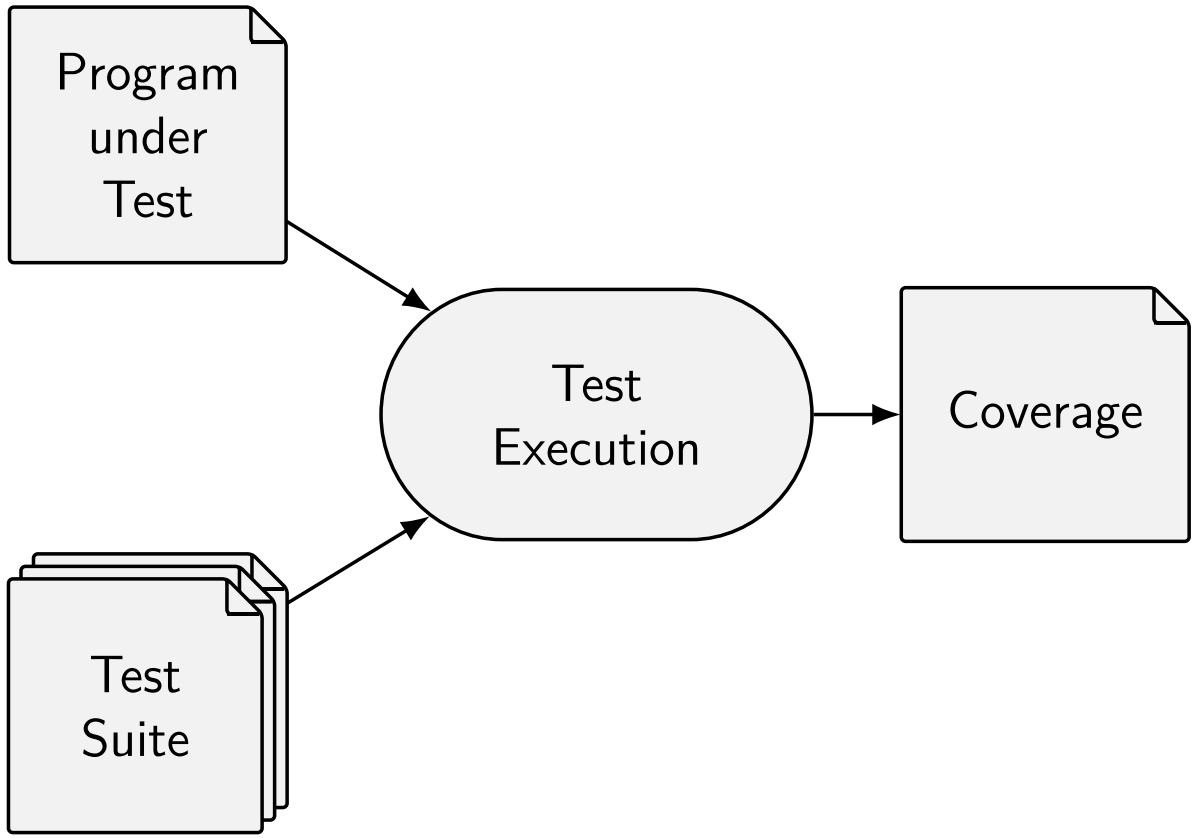
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

2022-04-04, Test-Comp 2022

SoSy-Lab Software Systems

Test-Comp 2021 talk:
https://youtu.be/pvePmeSJEwM

Test-Comp Scoring



The Issue

```
1  #include <stdio.h>
2  #include <unistd.h>
3  extern char input();
4
5  int main() {
6    char x = input();
7    if (x == 'a') {
8      while (1) {
9          fork();
10     }
11   } else {
12     remove("important.txt");
13     if (access("important.txt", F_OK) != -1) {
14       return 1;
15     }
16   }
17 }
```

The Issue

```
1  #include <stdio.h>
2  #include <unistd.h>
3  extern char input();
4
5  int main() {
6    char x = input();
7    if (x == 'a') {
8      while (1) {
9        fork();
10     }
11   } else {
12     remove("important.txt");
13     if (access("important.txt", F_OK) != -1) {
14       return 1;
15     }
16   }
17 }
```

The Issue

```
1  #include <stdio.h>
2  #include <unistd.h>
3  extern char input();
4
5  int main() {
6    char x = input();
7    if (x == 'a') {
8      while (1) {
9        fork();
10     }
11   } else {
12     remove("important.txt");
13     if (access("important.txt", F_OK) != -1) {
14       return 1;
15     }
16   }
17 }
```

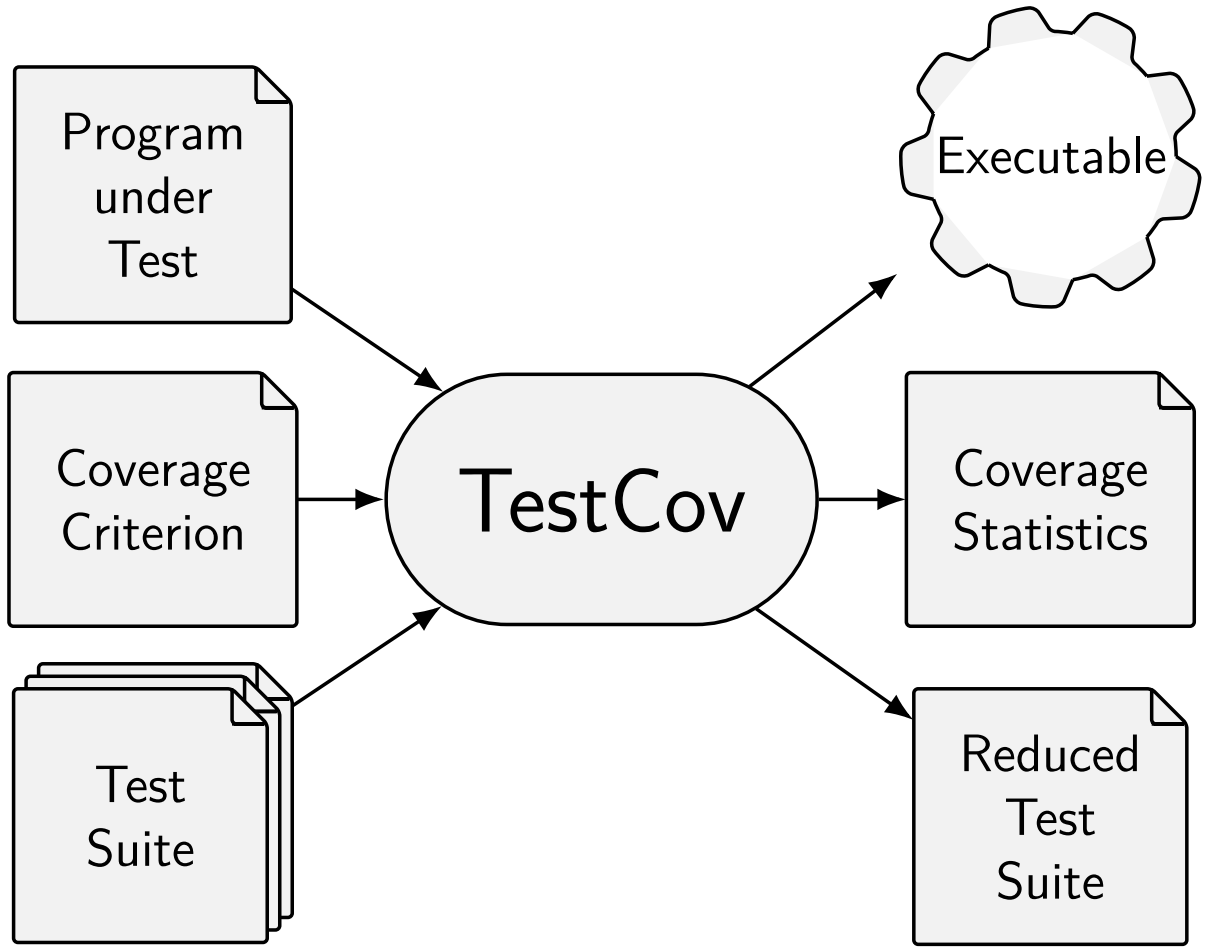
The Issue

```
1  #include <stdio.h>
2  #include <unistd.h>
3  extern char input();
4
5  int main() {
6    char x = input();
7    if (x == 'a') {
8      while (1) {
9        fork();
10     }
11   } else {
12     remove("important.txt");
13     if (access("important.txt", F_OK) != -1) {
14       return 1;
15     }
16   }
17 }
```

- Goal: Achieve 100 % branch coverage
- But: We don't want to use our systems to execute a test suite that achieves that.

Our Solution

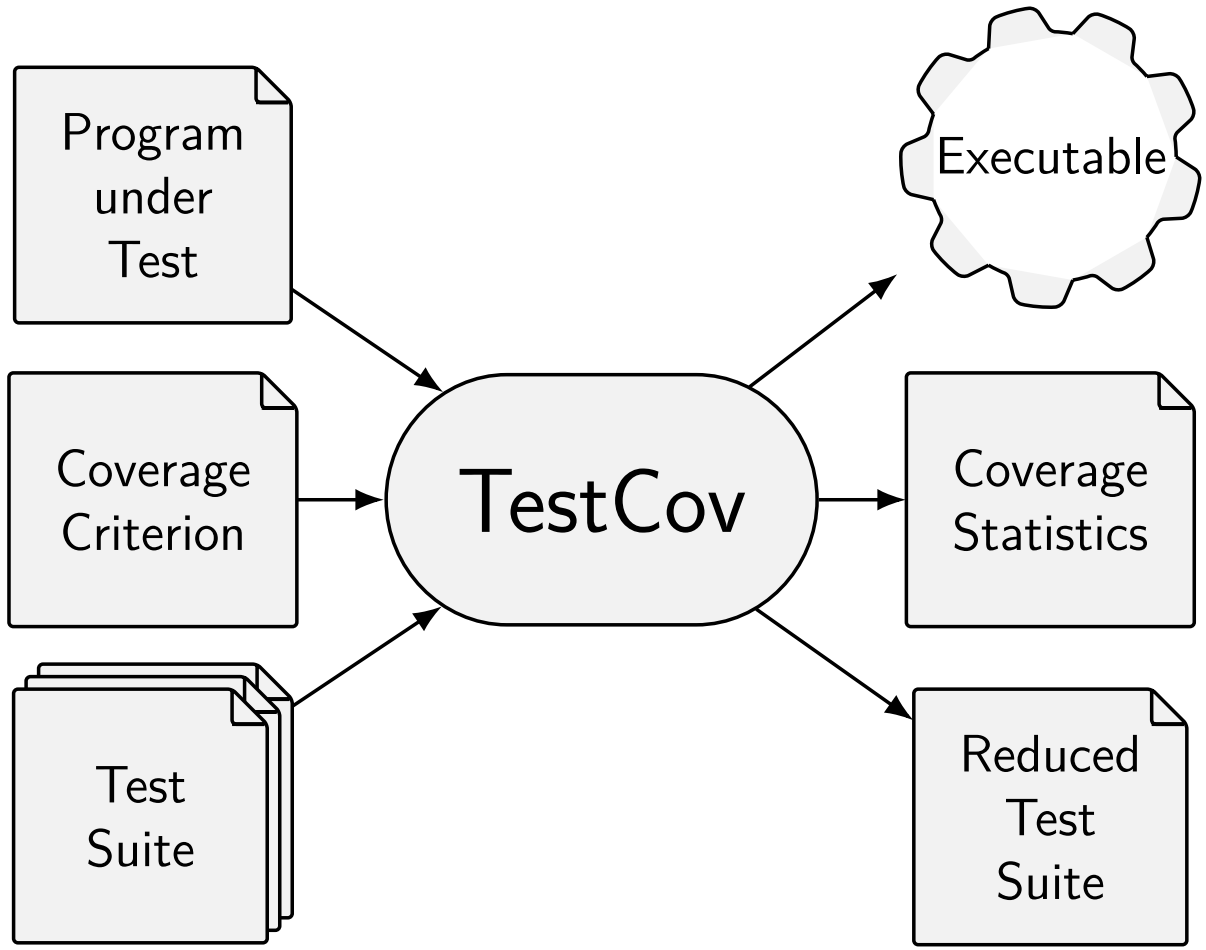
- Test isolation through Linux kernel features
- Coherent, single tool (for C programs)



Coverage Measurements

- Measurement through `lcov` and `llvm-cov` or `gcov`
  - Provide line- and condition-coverage
  - Unfitting definition of branch-coverage
- Branch coverage manually computed through program instrumentation

- Produced data:
  - Test success
  - Individual test coverage
  - Accumulated test coverage (after each execution)
  - Individual resource measurements (as JSON, plot)


Program
under
Test
Executable
Coverage
Criterion
TestCov
Coverage
Statistics
Test
Suite
Reduced
Test
Suite

News

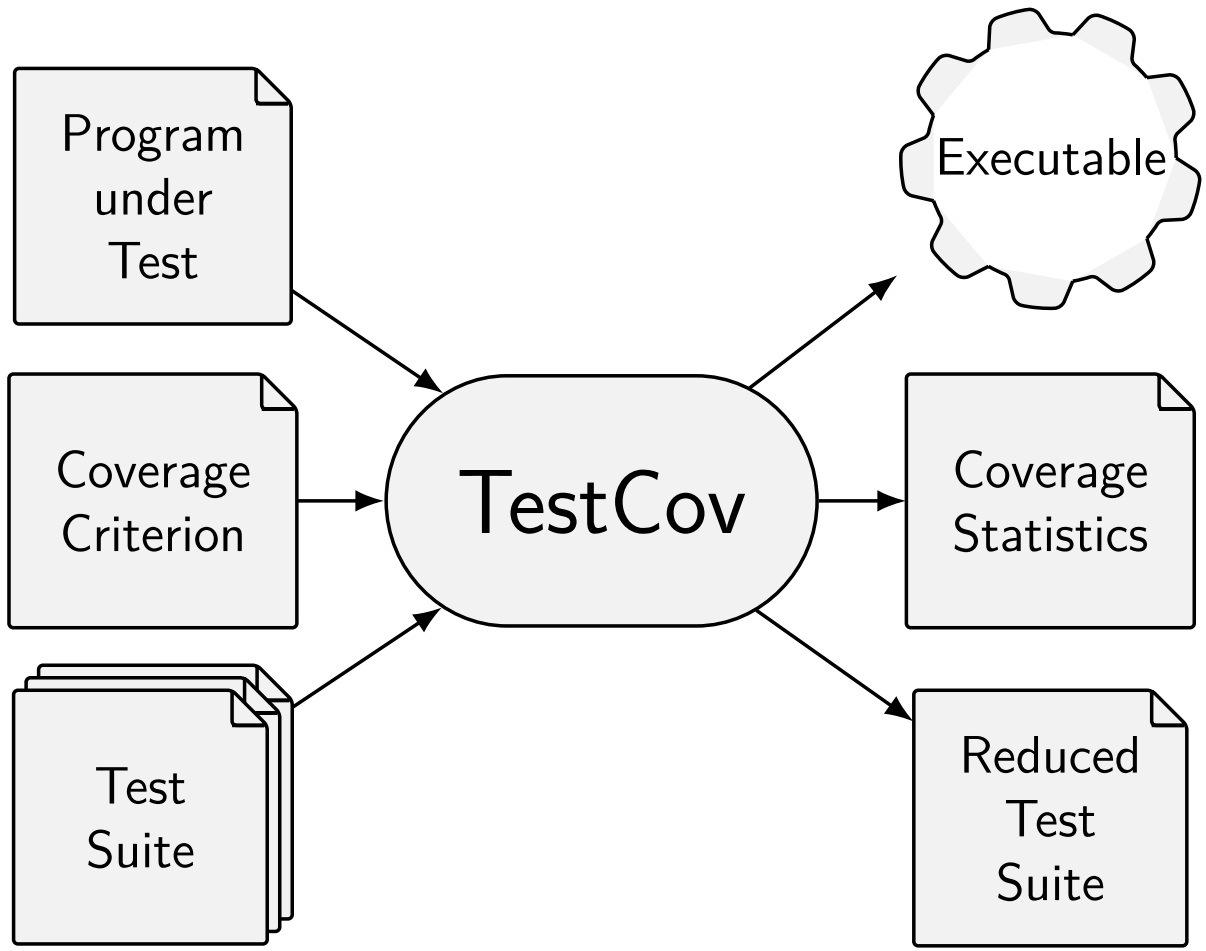
- Coverage-measurement reimplementation:
  Replaced pycparser with clang libtooling
- Measurement recording on aborts

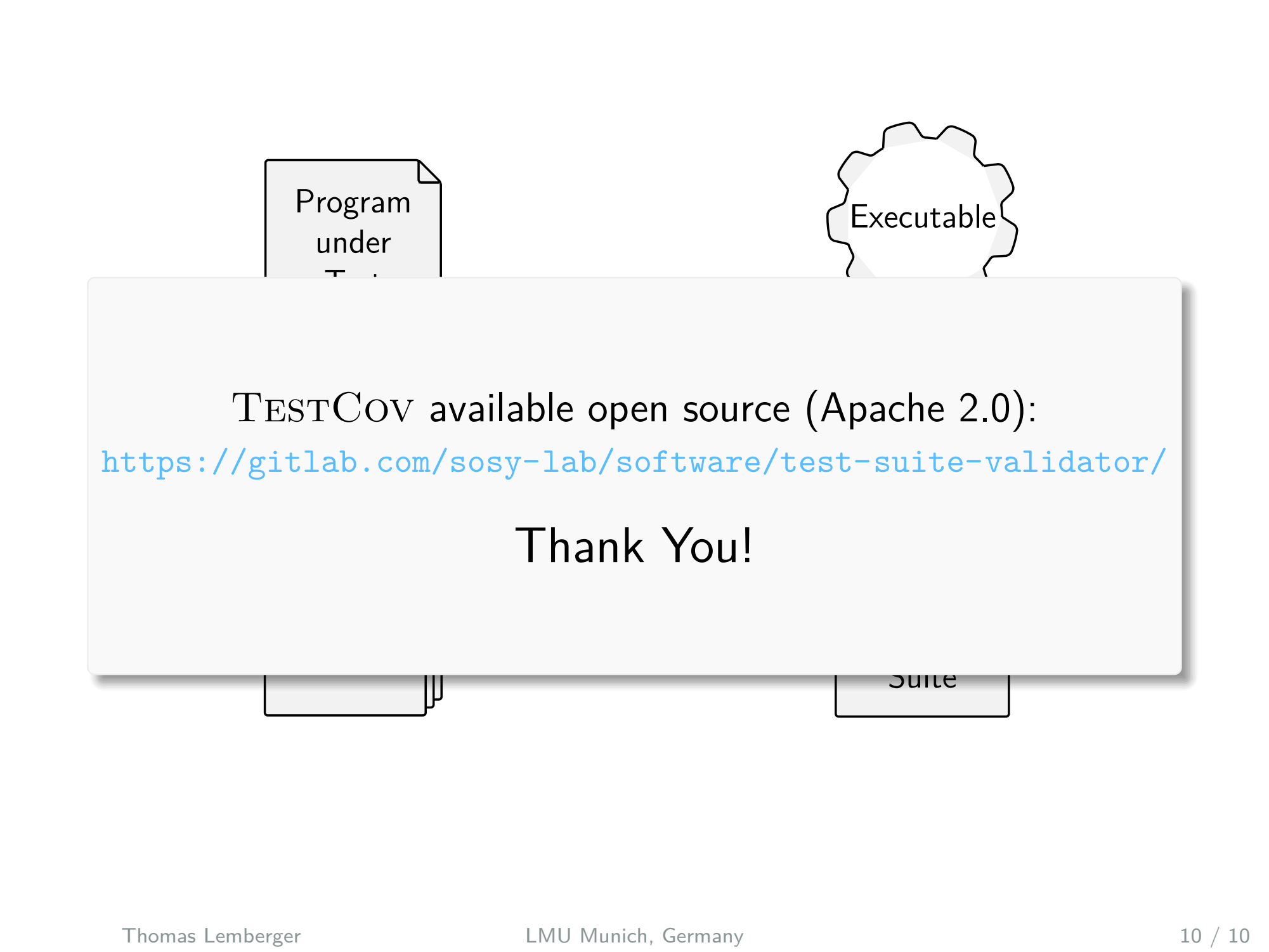
$\Rightarrow$ Improved speed, precision, full C language support

$\Rightarrow$ Support SQLite and all BusyBox test-generation tasks

Working with TestCov

- Updated time limit per single test execution: 50 s (before: 3 s)
- Memory limit per single test execution: 6 GB
- All produced data hidden in Test-Comp tables: Output files, reduced test suite
- Local usage without CGroup setup: ––no–isolation
  - Dangerous
  - No CPU time and memory measurement


Program under Test
Coverage Criterion
Test Suite
TestCov
Executable
Coverage Statistics
Reduced Test Suite



TestCov available open source (Apache 2.0):

https://gitlab.com/sosy-lab/software/test-suite-validator/

Thank You!

Test-Suite Format

- XML-based
- Two components:
  1. `metadata.xml`
  2. one XML-file per test case
     - Sequence of test inputs
- Handled as zip archive

Metadata

```
<?xml version="1.0"?>
<!DOCTYPE test-metadata PUBLIC "+//IDN sosy-lab.org//DTD test-format te
<test-metadata>
  <sourcecodelang>C</sourcecodelang>
  <producer>Testsuite Validator v2.0</producer>
  <specification>CHECK(FQL(cover EDGES(@CONDITIONEDGE)))</specificatio
  <programfile>example.c</programfile>
  <programhash>eeecda9cbf27c43c9017fa00dd900c19a5ec18d46303f59a6e0357db78
  <entryfunction>main</entryfunction>
  <architecture>32bit</architecture>
  <inputtestsuitefile>original-suite.zip</inputtestsuitefile>
  <inputtestsuitehash>11911d658dcfbf8501390bf0faa96eb193b11bb1</inputtestsui
  <creationtime>2019-06-19T14:17:34Z</creationtime>
</test-metadata>
```

Test Case

```
<?xml version="1.0"?>
<!DOCTYPE testcase PUBLIC "+//IDN sosy−lab.org//DTD test−format testcase
<testcase>
  <input>'b'</input>
  <input>10</input>
  <input>0x0f</input>
</testcase>
```