

Including mutation testing as part of a continuous integration workflow

Tim Waterson

What is mutation testing?

If our application code breaks, will our tests detect that?

```
def flip(char):  
    codepoint = ord(char)  
  
    if 64 < codepoint < 91:  
        return chr(codepoint + 32)  
  
    if 96 < codepoint < 123:  
        return chr(codepoint - 32)  
  
    return char
```

► 'a' → 'A'

► 'z' → 'Z'

► '1' → '1'

► 'Z' → 'z'

► 'A' → 'a'

► '+' → '+'

ROR Relational Operator Replacement

This mutation operator swaps relational operators. Substitution rules implemented by this operator (**Table 4**) refer to similar relational operators (eg. `<` vs `<=`) or opposite ones (eg. `<` vs `>`).

Table 4. Substitution rules of relational operators

Operator before mutation	<	<	>	>	<=	<=	>=	>=	==	!=
Operator after mutation	>	<=	<	>=	>=	<	<=	>	!=	==

Examples before mutation:

```
if x < y:  
while i >= 10:
```

after mutation:

```
if x > y:  
while i > 10:
```

Figure 1: The ROR mutation operator. Image is an extract from *Operators for Mutation Testing of Python Programs*, a 2014 paper by Derezinska and Halas.

```
def flip(char):  
    codepoint = ord(char)  
  
    if 64 < codepoint <= 91:  
        return chr(codepoint + 32)  
  
    if 96 < codepoint < 123:  
        return chr(codepoint - 32)  
  
    return char
```

All our tests still pass. . .

- ▶ 'a' → 'A'
- ▶ 'z' → 'Z'
- ▶ '1' → '1'
- ▶ 'Z' → 'z'
- ▶ 'A' → 'a'
- ▶ '+' → '+'

. . . however we now have an erroneous result.

- ▶ '[' → '{'


```
1. bash
[tim@tims-mac caseflip]$ mut.py --target caseflip --unit-test "caseflip-test"
[*] Start mutation process:
  - targets: caseflip
  - tests: caseflip-test
[*] 7 tests passed:
  - caseflip-test [0.00025 s]
[*] Start mutants generation and execution:
  - [# 1] AOR caseflip:8 : [0.01315 s] killed by testUppercaseA (caseflip-test.TestFlippingCharacters)
  - [# 2] AOR caseflip:11 : [0.00540 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 3] COI caseflip:7 : [0.00552 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 4] COI caseflip:10 : [0.00513 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 5] CRP caseflip:2 : [0.00555 s] killed by testAWholeString (caseflip-test.TestFlippingStrings)
  - [# 6] CRP caseflip:7 : [0.00569 s] killed by testUppercaseA (caseflip-test.TestFlippingCharacters)
  - [# 7] CRP caseflip:7 : [0.00559 s] survived
  - [# 8] CRP caseflip:8 : [0.00535 s] killed by testUppercaseA (caseflip-test.TestFlippingCharacters)
  - [# 9] CRP caseflip:10 : [0.00578 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 10] CRP caseflip:10 : [0.00470 s] survived
  - [# 11] CRP caseflip:11 : [0.00499 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 12] ROR caseflip:7 : [0.00613 s] killed by testNumberOne (caseflip-test.TestFlippingCharacters)
  - [# 13] ROR caseflip:7 : [0.00580 s] killed by testAWholeString (caseflip-test.TestFlippingStrings)
  - [# 14] ROR caseflip:7 : [0.00529 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 15] ROR caseflip:7 : [0.00538 s] survived
  - [# 16] ROR caseflip:10 : [0.00528 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 17] ROR caseflip:10 : [0.00524 s] survived
  - [# 18] ROR caseflip:10 : [0.00534 s] killed by testLowercaseA (caseflip-test.TestFlippingCharacters)
  - [# 19] ROR caseflip:10 : [0.00490 s] survived
[*] Mutation score [0.19915 s]: 73.7%
  - all: 19
  - killed: 14 (73.7%)
  - survived: 5 (26.3%)
  - incompetent: 0 (0.0%)
  - timeout: 0 (0.0%)
```

Figure 2: Mutation testing output from MutPy

What is continuous integration?

Every time the code changes, build the system and check its quality.



Figure 3: One way to show the CI build status

The screenshot shows a web browser window displaying the Travis CI build page for the `mutiny / mutiny` project. The browser's address bar shows the URL `https://travis-ci.org/mutiny/mutiny/builds/76785330`. The Travis CI logo and navigation links (Blog, Status, Help) are at the top. A green button for "Sign in with GitHub" is visible. Below the header, a message encourages users to "Help make Open Source a better place and start building better software today!". The main content area shows the build status as "build passing" in a green box. Navigation tabs include "Current", "Branches", "Build History", "Pull Requests", and "Build #86". A "Settings" dropdown menu is also present. The build details section shows a green checkmark and the message "Prefer instance_double so that doubles are verified." It also lists the commit hash "Commit 47a91c8", the comparison "Compare 149640d..47a91c8", and the author "Louis Rose authored and committed". On the right, it indicates "#86 passed", "Elapsed time 2 min 4 sec", and "4 months ago". Below this, a terminal window displays the build log, which includes system information, git clone command, rvm installation, bundle installation, and ruby version output. A "Download log" button is located at the top right of the terminal window.

Build #86 - mutiny/mutiny x

https://travis-ci.org/mutiny/mutiny/builds/76785330

Travis CI Blog Status Help

Sign in with GitHub

Help make Open Source a better place and start building better software today!

mutiny / mutiny build passing

Current Branches Build History Pull Requests > Build #86 Settings

✓ Prefer instance_double so that doubles are verified.

Commit 47a91c8

Compare 149640d..47a91c8

Louis Rose authored and committed

↔ #86 passed

Elapsed time 2 min 4 sec

4 months ago

Download log

```
1 Using worker: worker-linux-da6418d7-2.bb.travis-ci.org:travis-linux-1
2
3 Build system information
67
68 $ git clone --depth=50 --branch=master git://github.com/mutiny/mutiny.git mutiny/mutiny
69 $ rvm use 2.2.2 --install --binary --fuzzy
70 $ export BUNDLE_GEMFILE=$PWD/Gemfile
103 $ ruby --version
104 ruby 2.2.2p95 (2015-04-13 revision 50295) [x86_64-linux]
105 $ rvm --version
106 rvm 1.26.10 (latest-minor) by Wayne E. Seguin <wayneesequin@gmail.com>, Michal Papis <mpapis@gmail.com> [https://rvm.io/]
107 $ bundle --version
```

Figure 4: Travis CI in action on a GitHub project

JUnit Lambda #209 Console

https://junit.ci.cloudbees.com/job/JUnit_Lambda/209/console

Builds Ecosystem Support Documentation Log In Sign Up

Jenkins JUnit Lambda #209

Previous Build Next Build

Executed Gradle Tasks

- documentation:clean
- unit-commons:clean
- unit-console:clean
- unit-engine-api:clean
- unit-engine:clean
- unit-launcher:clean
- unit-tests:clean
- unit4-engine:clean
- unit4-launcher-runner:clean
- unit5-api:clean
- unit5-engine:clean
- sample-extension:clean
- sample-project:clean
- surefire-junit5:clean
- documentation:spotlessGroovyCheck
- documentation:spotlessJavaCheck

```

> git rev-list 554912b8f895d7ede966af64fecd6b392eb3072 # timeout=10
[Gradle] - Launching build.
[JUnit_Lambda] $ /scratch/jenkins/workspace/JUnit_Lambda/gradlew --gradle-user-
home=/scratch/jenkins/workspace/JUnit_Lambda/.gradle clean check assemble aggregateJavadocs uploadArchives
publishGhPages
:documentation:clean
:unit-commons:clean
:unit-console:clean
:unit-engine-api:clean
:unit-engine:clean
:unit-gradle:clean
:unit-launcher:clean
:unit-tests:clean
:unit4-engine:clean
:unit4-launcher-runner:clean
:unit5-api:clean
:unit5-engine:clean
:sample-extension:clean
:sample-project:clean
:surefire-junit5:clean
:documentation:spotlessGroovyCheck
:documentation:spotlessJavaCheck FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':documentation:spotlessJavaCheck'.
> Format violations were found. Run 'gradlew spotlessApply' to fix them.
   junit5-engine/src/main/java/org/junit/gen5/engine/junit5/execution/TestExtensionRegistry.java

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output.

BUILD FAILED

Total time: 47.99 secs
Build step 'Invoke Gradle script' changed build result to FAILURE
Build step 'Invoke Gradle script' marked build as failure
Recording test results
ERROR: Publisher 'Publish JUnit test result report' failed: No test report files were found. Configuration error?
Publishing Javadoc
Finished: FAILURE

```

Contact Us Privacy Policy Terms of Service © 2015 CloudBees, Inc.

Figure 5: A failed build in the Jenkins console

What are the benefits arising from bringing mutation testing and continuous integration together?

What are the challenges in running mutation testing in a continuous integration environment?

Speed

Reporting

What's my project doing to address those challenges?

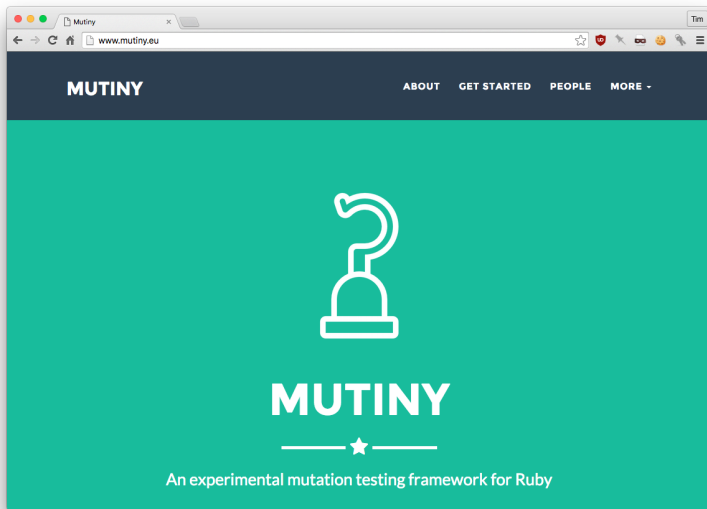


Figure 6: The mutiny website

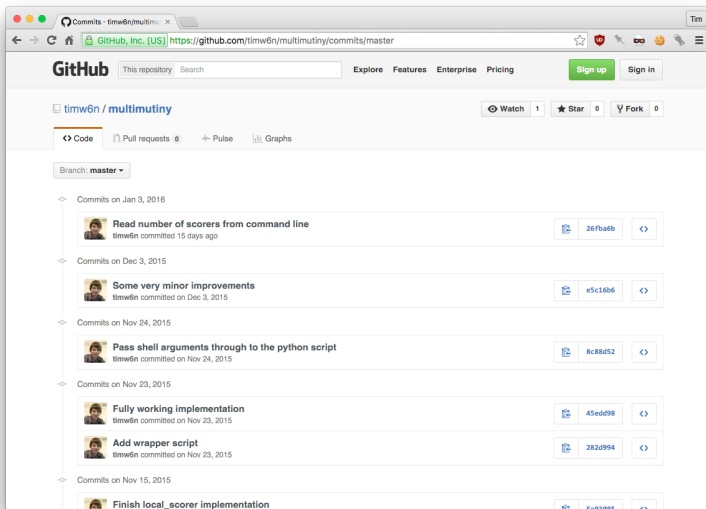


Figure 7: multimutiny, the main codebase for my project

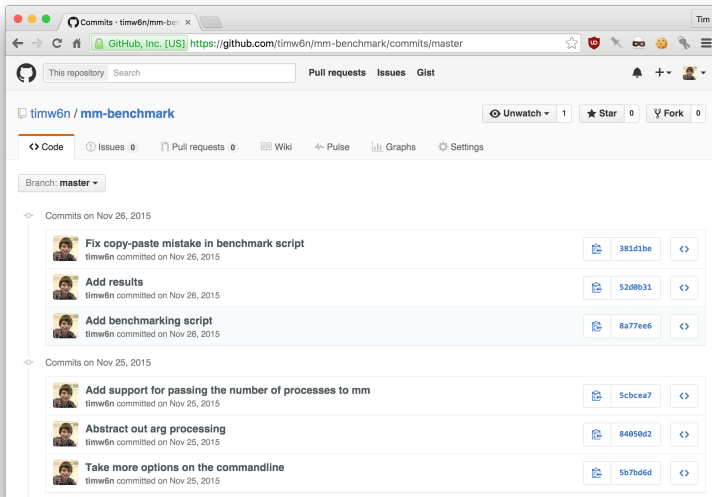


Figure 8: mm-benchmark, a tool for comparing multimutiny and mutiny performance across different-sized codebases and different numbers of parallel processes

Any questions?