Make your tests fail
How machines can help find corner cases that humans ignore

Isabel Drost-Fromm
Isabel Drost-Fromm

Software Engineer Elasticsearch
Member Apache Software Foundation
Co-founder Apache Mahout
Co-founder Berlin Buzzwords

“Not good enough an engineer to write code w/o tests.”
public void testReclaimedDeletes() throws Exception {
    setIndexWriterMaxDocs(10);
    try {
        Directory dir = newDirectory();
        IndexWriter w = new IndexWriter(dir, new IndexWriterConfig(null));
        for (int i = 0; i < 10; i++) {
            Document doc = new Document();
            doc.add(newStringField("id", "\"" + i + \"", Field.Store.NO));
            w.addDocument(doc);
        }
        // Delete 5 of them:
        for (int i = 0; i < 5; i++) {
            w.deleteDocuments(new Term("id", "\"" + i + \""));
        }
        w.forceMerge(1);
        assertEquals(5, w.maxDoc());
        // Add 5 more docs
        for (int i = 0; i < 5; i++) {
            w.addDocument(new Document());
        }
        // 11th document should fail:
        try {
            w.addDocument(new Document());
        } catch (IOException ioe) {
            // Expected
        }
    } finally {
        // Close the writer
        w.close();
    }
}
You regularly check code coverage?
You run your tests regularly?
Java:

```java
int input = generateInteger(
    Integer.MIN_VALUE,
    Integer.MAX_VALUE);

int output = Math.abs(input);
```
Java:

```java
int input = generateInteger(
    Integer.MIN_VALUE,
    Integer.MAX_VALUE);

int output = Math.abs(input);

output == -2147483648
```
carrotsearch randomized testing
Input data
- numbers
- strings
- specific charset only

Environment
- change
- Locale
Input data
- numbers
- strings
- specific charset only

Environment
- change
- Locale

Reproducible due to specific seed.

Repeating tests runs them on different data.

Support for test timeouts.

Detects leaking threads.

Annotate tests to e.g. run only nightly.
Checking result cheaper than computing it.
Use slower but simpler algorithm for comparison.
Cheap upper/lower bound on result.
@Test
@Repeat (iterations = 100)
public void testSorting {
    int length = randomIntBetween(10, 100);
    short[] list = new short[length];
    for (int i = 0; i < length; i++) {
        list[i] = randomShort();
    }
    short[] result = Arrays.sort(list);
    assertTrue(isSorted(result));
}
@ThreadLeakScope(Scope.TEST)
@ThreadLeakAction({Action.WARN, Action.INTERRUPT})
@ThreadLeakLinger(linger = 1000)
public static class TheGoodBadAndUgly extends RandomizedTest {

    @Test
    public void good() {
        // I do nothing and I'm good.
    }

    @Repeat(iterations = 10)
    @Test
    public void bad() {
        // I fail randomly, about 20% of the time.
        assertFalse(randomIntBetween(1, 100) <= 20);
    }

    @Test
    public void ugly() {
        // I start and abandon a thread which falls
        // outside the test's scope. The test will fail.
        new Thread() {
            public void run() {
                RandomizedTest.sleep(5000);
            }
        }.start();
    }

    @Seed("AF567B2B9F8A9F1C")
    @BeforeClass
    public static void printMasterContext() {
        System.out.println("# Static context
        System.out.println(getContext().getR:}

Copyright Elasticsearch 2015.
JUnit Test Results:

- Total Tests: 12
- Errors: 1
- Failures: 1

Errors:
- com.carrotsearch.randomizedtesting.snippets:
  - bad: #0 seed=[6C6E22495E089CC9:D9F
  - bad: #1 seed=[6C6E22495E089CC9:6D
  - bad: #2 seed=[6C6E22495E089CC9:E34
  - bad: #3 seed=[6C6E22495E089CC9:D2
  - bad: #4 seed=[6C6E22495E089CC9:9E6
  - bad: #5 seed=[6C6E22495E089CC9:F9E
  - bad: #6 seed=[6C6E22495E089CC9:314
  - bad: #7 seed=[6C6E22495E089CC9:AD
  - bad: #8 seed=[6C6E22495E089CC9:9F5
  - bad: #9 seed=[6C6E22495E089CC9:48E

- ugly: 11.622 s
REPRODUCE WITH: gradle :core:integTest -Dtests.seed=6CB9C3FB4D9E4867
-Dtests.class=org.elasticsearch.indices.mapping.UpdateMappingIntegrationIT
-Dtests.method="testDynamicUpdates"
-Des.logger.level=WARN
-Dtests.security.manager=true
-Dtests.locale=es_CL
-Dtests.timezone=America/Jamaica
On the unit test level:

Random input w/ fixed seed to increase search space.

In CI re-running tests covers adds to coverage increase.
This machine is a ser
DO NOT POWE
... DOWN!!
For distributed systems:

Vary the number of nodes.

Vary the number of processes on a node.
For systems running on the JVM:

Vary JVM optimisation parameters.

Vary the JVM distribution used for testing.
On commit

Smoke test

Java unit and integration tests

REST tests

Runs continuously

Coverage

Platform specific test (EC2 and bare metal)

- Windows
- Ubuntu
- Debian
- CentOS
Who broke the build?
Uncover bugs in your environment.
GOTOber - where's your testing pain?