



David Pilato

Developer | Evangelist

@dadoonet

Le hasard fait bien les tests



Breizh C@mp
La conférence à l'Ouest

Vous écrivez des tests ?

Idéalement avant d'écrire le code

```
public void testReclaimedDeletes() throws Exception {
    setIndexWriterMaxDocs(10);
    try {
        Directory dir = newDirectory();
        IndexWriter w = new IndexWriter(dir, new IndexWriterConfig());
        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 {
```

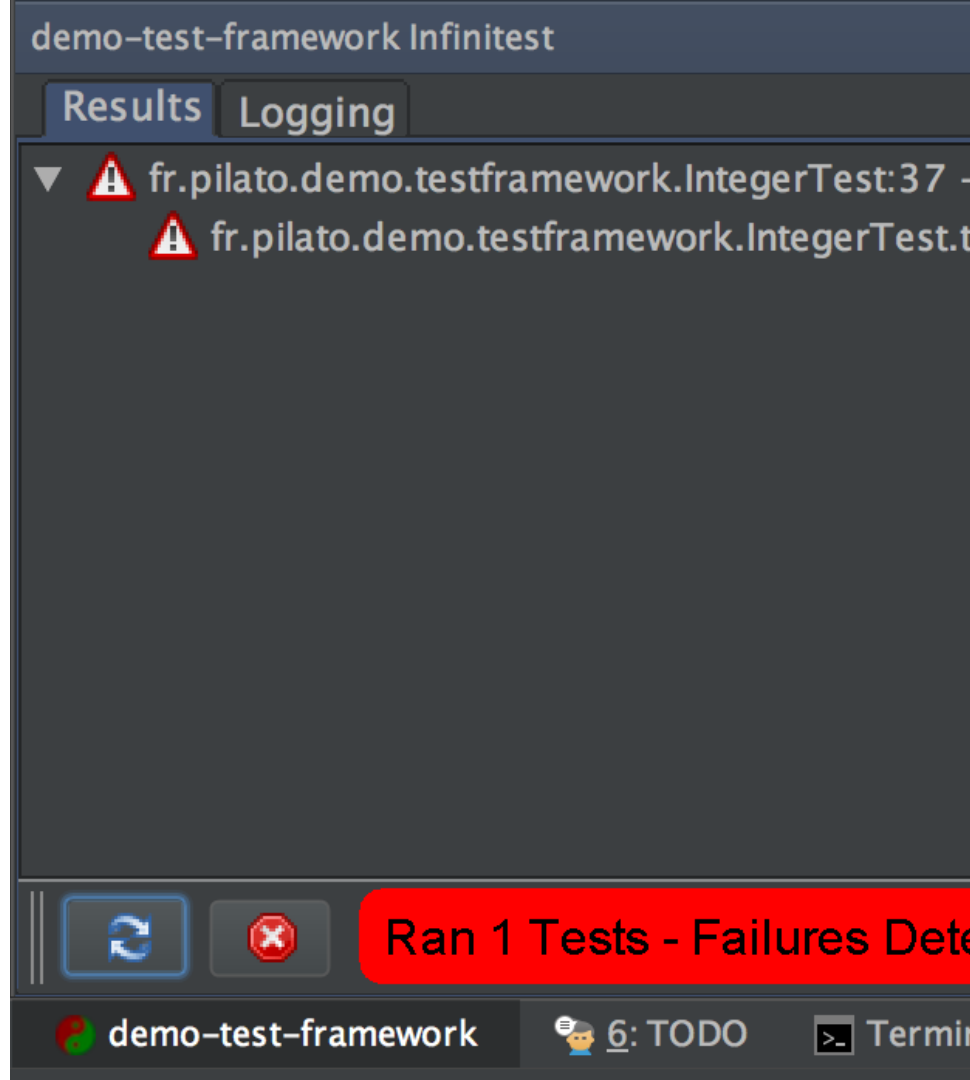

Couverture des tests

Vous vérifiez ?

743	9417		<code>public IndexWriter(Directory d, IndexW</code>
744	9417		<code>conf.setIndexWriter(this); // prevent</code>
745	9416		<code>config = conf;</code>
746	9416		<code>directory = d;</code>
747	9416		<code>analyzer = config.getAnalyzer();</code>
748	9416		<code>infoStream = config.getInfoStream();</code>
749	9416		<code>mergeScheduler = config.getMergeSched</code>
750	9416		<code>codec = config.getCodec();</code>
751			
752	9416		<code>bufferedUpdatesStream = new BufferedU</code>
753	9416		<code>poolReaders = config.getReaderPooling</code>
754			
755	9416		<code>writeLock = directory.makeLock(WRITE_</code>
756			
757	9414		<code>if (!writeLock.obtain(config.getWrite</code>
758	0		<code>throw new LockObtainFailedException</code>
759			
760	9414		<code>boolean success = false;</code>
761	9414		<code>try {</code>
762	9414		<code>OpenMode mode = config.getOpenMode(</code>
763	9414		<code>boolean create;</code>
764	9414		<code>if (mode == OpenMode.CREATE) {</code>
765	1963		<code>create = true;</code>
766	7451		<code>} else if (mode == OpenMode.APPEND)</code>
767	802		<code>create = false;</code>
768			<code>} else {</code>

Infinittest

En continu ?



CI

Vous exécutez les tests
régulièrement ?

Jenkins Lucene-Artifacts-5.x

 [Back to Dashboard](#)

 [Status](#)

 [Changes](#)

 [Dependency Graph](#)

 [Document](#)

 [Email Template Testing](#)

 **Build History** (trend) 

	#710	Dec 15, 2014 8:37:25 AM
	#709	Dec 13, 2014 9:51:25 PM
	#708	Dec 12, 2014 11:11:03 PM
	#707	Dec 11, 2014 3:16:33 PM
	#706	Dec 10, 2014 10:27:58 PM
	#705	Dec 9, 2014 11:14:25 PM
	#704	Dec 8, 2014 3:55:42 AM

Project L

 [Do](#)

 [La](#)

 [Re](#)

Upstream F

 [Lucen](#)

Downstream

 [Solr-A](#)

Permalinks



Java

```
public static int generateInteger(int min, int max) {  
    return ThreadLocalRandom.current().nextInt(min, max);  
}  
  
@Test  
public void testInteger() {  
    int input = generateInteger(  
        Integer.MIN_VALUE,  
        Integer.MAX_VALUE);  
    int output = Math.abs(input);  
  
    assertThat(output, greaterThanOrEqualTo(0));  
}
```

Results Logging

▼  fr.pilato.demo.testframework.IntegerTest:37 – AssertionError(Expected: a value equal to or greater than <0> but: <-2147483648> was less than <0>)
 fr.pilato.demo.testframework.IntegerTest.testInteger

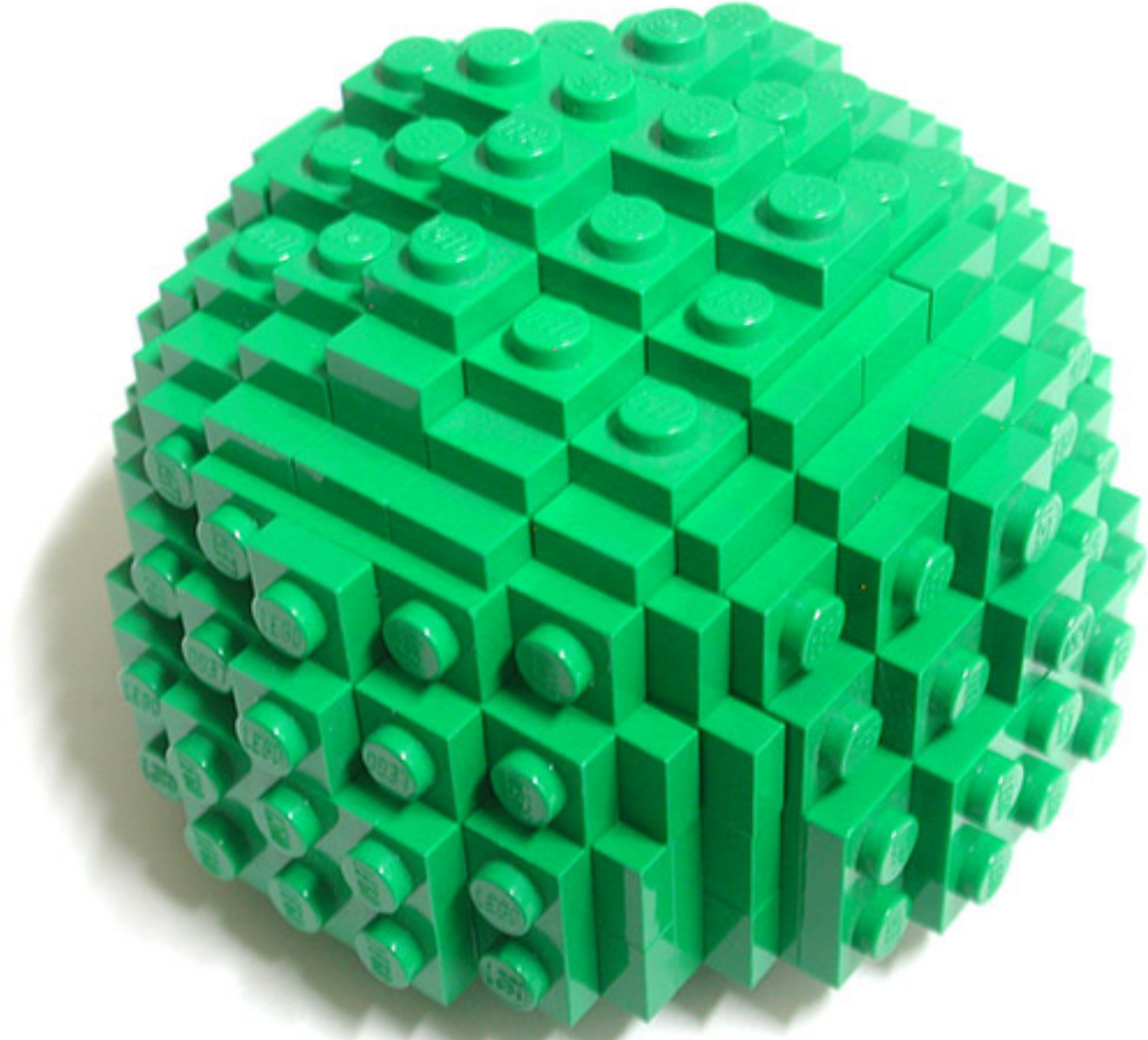


Ran 0 Tests - Failures Detected

Java

WTF?

```
public static int generateInteger(int min, int max) {  
    return ThreadLocalRandom.current().nextInt(min, max);  
}  
  
@Test  
public void testInteger() {  
    int input = generateInteger(  
        Integer.MIN_VALUE,  
        Integer.MAX_VALUE);  
    int output = Math.abs(input);  
  
    assertThat(output, greaterThanOrEqualTo(0));  
}
```







carrotsearch randomized testing

<https://github.com/randomizedtesting/>

Ajouter la dépendance au projet

pom.xml

```
<dependency>
  <groupId>com.carrotsearch.randomizedtesting</groupId>
  <artifactId>randomizedtesting-runner</artifactId>
  <version>2.3.3</version>
  <scope>test</scope>
</dependency>
```

Désactiver surefire

pom.xml

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.19</version>
  <executions>
    <execution>
      <id>default-test</id>
      <phase>none</phase>
    </execution>
  </executions>
</plugin>
```

Activer randomizedtesting

pom.xml

```
<plugin>
  <groupId>com.carrotsearch.randomizedtesting</groupId>
  <artifactId>junit4-maven-plugin</artifactId>
  <version>2.3.3</version>

  ...
</plugin>
```

Configurer randomizedtesting

pom.xml

```
...
<configuration>
  <heartbeat>10</heartbeat>
  <jvmOutputAction>pipe,ignore</jvmOutputAction>
  <leaveTemporary>true</leaveTemporary>
  <ifNoTests>warn</ifNoTests>
  <listeners>
    <report-text showThrowable="true" showStackTraces="true" showOutput="always"
      showStatusOk="true" showStatusError="true"
      showStatusFailure="true" showStatusIgnored="true"
      showSuiteSummary="true" />
  </listeners>

  <systemProperties combine.children="append">
    <arg.common>arg.common</arg.common>
  </systemProperties>
</configuration>
...
```

Configurer randomizedtesting pour unit tests

pom.xml

```
...
<executions>
  <execution>
    <id>unit-tests</id>
    <phase>test</phase>
    <goals>
      <goal>junit4</goal>
    </goals>
  </execution>
</executions>
```

Run

mvn test

```
[INFO] --- junit4-maven-plugin:2.3.3:junit4 (unit-tests) @ demo-test-framework ---  
[INFO] <JUnit4> says iHola! Master seed: AEF86C96467D2F4F  
Executing 2 suites with 1 JVM.
```

```
Started J0 PID(20619@MacBook-Pro-4.local).  
Suite: fr.pilato.demo.testframework.IntegerTest  
OK      0.01s | IntegerTest.testInteger  
Completed [1/2] in 0.02s, 1 test
```

```
Suite: fr.pilato.demo.testframework.RandomTest  
FAILURE 0.01s | RandomTest.testFail <<<  
  > Throwable #1: java.lang.AssertionError: fail the test  
  >   at __randomizedtesting.SeedInfo.seed([AEF86C96467D2F4F:235BBDF496C25F4E]:0)  
  >   at org.junit.Assert.fail(Assert.java:88)  
  >   at fr.pilato.demo.testframework.RandomTest.testFail(RandomTest.java:31)
```

```
...
```




Random ?

seed

```
@Test
public void testSeed() {
    Random generator = new Random();
    int num = generator.nextInt();
    assertThat(num, is(1553932502));
}
```

Random ?

seed

```
@Test
public void testSeed() {
    Random generator = new Random(12345L);
    int num = generator.nextInt();
    assertThat(num, is(1553932502));
}
```




Random test parameters

```
@Test
```

```
public void testInteger() {  
    int num = Math.abs(randomInt());  
    assertThat(num, greaterThanOrEqualTo(0));  
}
```

```
Suite: fr.pilato.demo.testframework.RandomTest
```

```
FAILURE 0.01s | RandomTest.testFail <<<
```

```
> Throwable #1: java.lang.AssertionError:  
> Expected: a value equal to or greater than <0>  
>    but: <-2147483648> was less than <0>  
>    at __randomizedtesting.SeedInfo.seed([AEF86C96467D2F4F:235BBDF496C25F4E]:0)  
>    at org.junit.Assert.fail(Assert.java:88)  
>    at fr.pilato.demo.testframework.RandomTest.testInteger(RandomTest.java:52)
```

On peut reproduire le test

en utilisant la seed

```
@Test
@Seed("AEF86C96467D2F4F")
public void testIntegerWithSeed() {
    int num = Math.abs(randomInt());
    assertThat(num, greaterThanOrEqualTo(0));
}
```

Rendre seed configurable

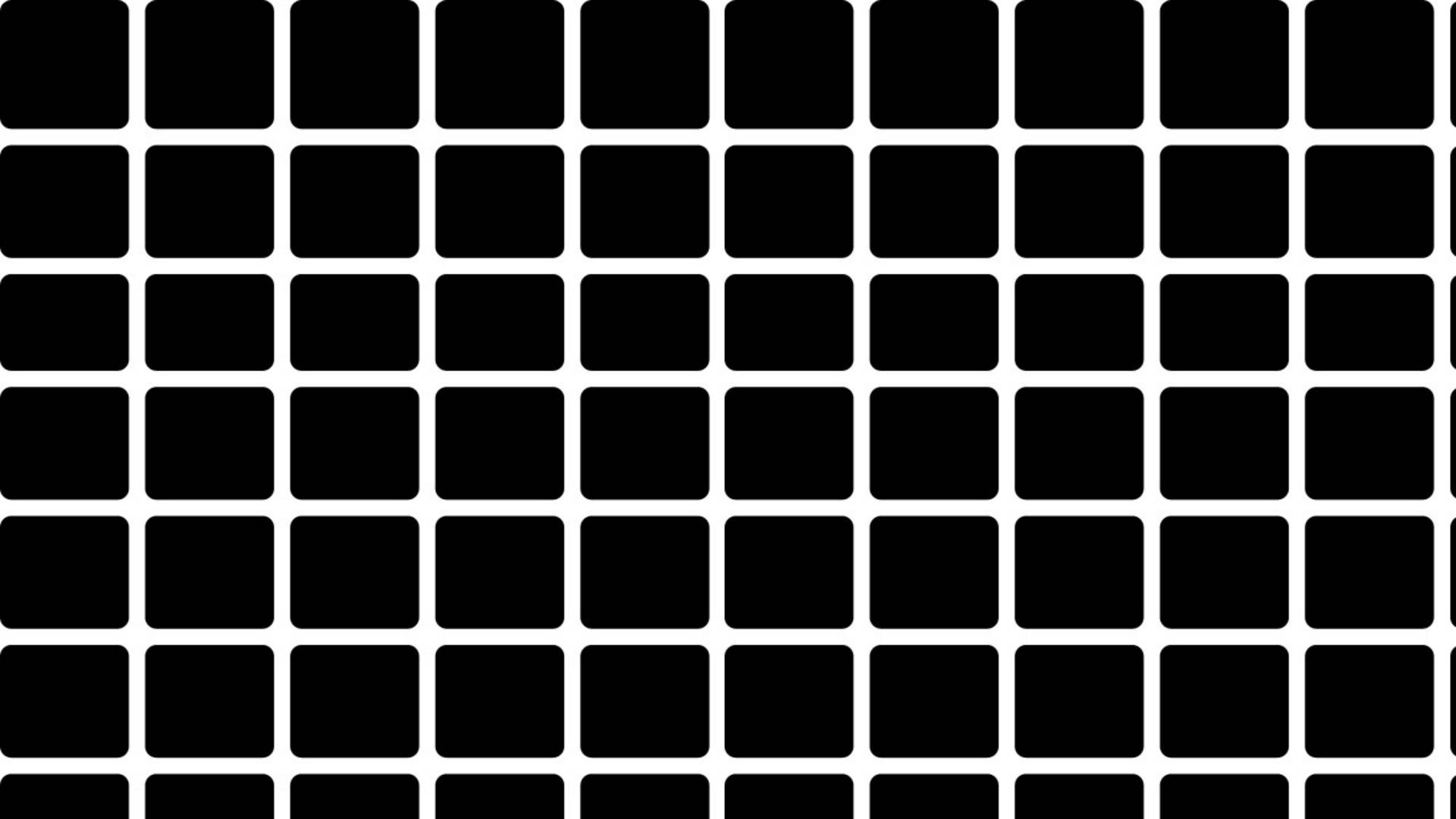
pom.xml

```
...
<configuration>
  <heartbeat>10</heartbeat>
  <jvmOutputAction>pipe,ignore</jvmOutputAction>
  <leaveTemporary>true</leaveTemporary>
  <ifNoTests>warn</ifNoTests>
  <listeners>
    <report-text showThrowable="true" showStackTraces="true" showOutput="always"
      showStatusOk="true" showStatusError="true"
      showStatusFailure="true" showStatusIgnored="true"
      showSuiteSummary="true" />
  </listeners>
  <seed>${tests.seed}</seed>
  <systemProperties combine.children="append">
    <arg.common>arg.common</arg.common>
  </systemProperties>
</configuration>
...
```

On peut reproduire le test depuis le CLI

en utilisant la seed

```
mvn test -Dtests.seed=AEF86C96467D2F4F
```

Changing the test context

- Input data
 - Numbers : `randomInt()`, `randomDouble()`, `between(1, 10)`, `atLeast(5)`, `atMost(10000)`...
 - Booleans : `randomBoolean()`
 - Strings : `randomAsciiOfLengthBetween(5, 30)`
 - TimeZones: `randomTimeZone()`
- Environnement
 - Locale
 - Charset

Playing with Locale

pom.xml

```
<properties>
  <tests.locale>random</tests.locale>
</properties>

<configuration>
  ...
  <systemProperties combine.children="append">
    <arg.common>arg.common</arg.common>
    <tests.locale>${tests.locale}</tests.locale>
  </systemProperties>
</configuration>
```

Playing with Locale

setup

```
private static final Locale savedLocale = Locale.getDefault();

@BeforeClass
public static void setLocale() {
    String testLocale = System.getProperty("tests.locale", "random");
    Locale locale = testLocale.equals("random") ? randomLocale() :
        new Locale.Builder().setLanguageTag(testLocale).build();
    Locale.setDefault(locale);
}

@AfterClass
public static void resetLocale() {
    Locale.setDefault(savedLocale);
}
```

Playing with Locale

launch tests

```
@Test
public void withLocale() {
    DateTimeFormatter dateTimeFormatter =
        DateTimeFormatter.ofLocalizedDate(FormatStyle.FULL);
    String format = dateTimeFormatter.format(LocalDate.now());
    System.out.println("locale = " + Locale.getDefault().toLanguageTag());
    System.out.println("format = " + format);
}
```

RUN 1

```
locale = es-PR
format = miércoles 9 de marzo de 2016
```

RUN 2

```
locale = tr-TR
format = 09 Mart 2016 Çarşamba
```

Playing with Locale

launch tests with a given Locale

```
mvn test -Dtests.locale=fr-FR
```

```
1> locale = fr-FR
```

```
1> format = mercredi 9 mars 2016
```





AEF86C96467D2F4F:235BBDF496C25F4E

2 seeds ?

AEF86C96467D2F4F:235BBDF496C25F4E

- AEF86C96467D2F4F : Contexte statique @BeforeClass @AfterClass
- 235BBDF496C25F4E : Contexte du test @Before @After

Lancer plusieurs fois le même test

histoire d'être vraiment certain !

```
@Test @Repeat(iterations = 10)
public void repeatMe() {
    Locale locale = randomLocale();
    DateTimeFormatter dateTimeFormatter =
        DateTimeFormatter.ofLocalizedDate(FormatStyle.FULL).withLocale(locale);
    String format = dateTimeFormatter.format(LocalDate.now());
    System.out.println("date is [" + format + "] with locale [" +
        locale.toLanguageTag() + "]);
}
```

Run RandomTest.repeatMe

Stopped. 10 of 21 tests passed - 1s 443ms

Test Name	Duration	Output
repeatMe	1s 443ms	/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home/bin/java ...
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:ABE85B69E20B2515]}	1s 66ms	date is [2016. ožujka 11] with locale [hr]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:1FBEE795D6C9EE39]}	52ms	date is [viernes 11 de marzo de 2016] with locale [es-CU]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:91577149870DA6F2]}	31ms	date is [viernes 11 de marzo de 2016] with locale [es-US]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:A0B9DAACEBF3FDD8]}	35ms	date is [11. mars 2016] with locale [no-NO]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:EC785F014AFB3D60]}	18ms	date is [piątek, 11 marca 2016] with locale [pl]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:7D828C5E37477060]}	18ms	date is [11 2016] with locale [ar-QA]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:435CE8D825776066]}	9ms	date is [Petek, 11. marec 2016] with locale [sl-SI]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:DDEF72A20663F4C8]}	56ms	date is [11 2016] with locale [ar-BH]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:ED4397CC71A8E392]}	91ms	date is [viernes 11 de marzo de 2016] with locale [es-CO]
RandomTest.repeatMe { seed=[BDFFFF96C4FD53B5:3AC8C17615FFD4C0]}	67ms	date is [piątek, 11 marca 2016] with locale [pl]

Process finished with exit code 0

De temps en temps...

ou souvent...

```
@Test
public void sometimes() {
    int bulkSize = randomIntBetween(500, 1000);

    for (int i = 0; i < bulkSize; i++) {
        addDocument("english_" + i, generatePerson());
        if (frequently()) {
            addDocument("french_" + i, generatePerson());
        }

        if (rarely()) {
            // Shutdown Node 1
            shutdownNode(1);
        }
    }
}
```

Ne pas lancer des tests inutiles

We know it will fail!

```
@Test
public void ignoreIfUseless() {
    boolean b = randomBoolean();

    assertThat(b, is(true));
}
```

Ne pas lancer des tests inutiles

We know it will fail!

```
@Test
public void ignoreIfUseless() {
    boolean b = randomBoolean();

    assumeTrue(b);
    assertThat(b, is(true));
}
```

Run RandomTest.ignoreIfUseless

Stopped. 1 of 9 tests ignored - 100ms

RandomTest (fr.pilato.demc 30ms

- sometimes
- testInteger
- ignoreIfUseless 30ms
- withLocale
- testNumbers
- testFail
- testIntegerWithSeed
- testSeed
- ignoreIfUseless 100ms

Process finished with exit code 0

t: <false>, expected: is <true>

[3769E797987F38B5:342256CF0A251A61]:0

at org.junit.Assume.assumeThat(Assume.java:95)
at org.junit.Assume.assumeTrue(Assume.java:41)
at com.carrotsearch.randomizedtesting.RandomizedTest.assumeTrue(RandomizedTest.java:597)
at fr.pilato.demc.testframework.RandomTest.ignoreIfUseless(RandomTest.java:141) <4 internal calls>
at com.carrotsearch.randomizedtesting.RandomizedRunner.invoke(RandomizedRunner.java:1764)
at com.carrotsearch.randomizedtesting.RandomizedRunner\$8.evaluate(RandomizedRunner.java:871)
at com.carrotsearch.randomizedtesting.rules.StatementAdapter.evaluate(StatementAdapter.java:36)
at com.carrotsearch.randomizedtesting.ThreadLeakControl\$StatementRunner.run(ThreadLeakControl.java:367)
at com.carrotsearch.randomizedtesting.ThreadLeakControl.forkTimeoutingTask(ThreadLeakControl.java:809)

En faire moins le jour...

```
@Test @Nightly
public void longRunningTest() {
    int bulkSize = randomIntBetween(500, 1000);

    for (int i = 0; i < bulkSize; i++) {
        try {
            Thread.sleep(Math.abs(between(100, 10000)));
        } catch (InterruptedException e) {
            assumeNoException(e);
        }
    }
}

$ mvn test

IGNOR/A 0.00s | RandomTest.longRunningTest
> Assumption #1: 'nightly' test group is disabled (@Nightly(value=))
```

Et plus la nuit !

```
<systemProperties combine.children="append">  
  <arg.common>arg.common</arg.common>  
  <tests.nightly>${tests.nightly}</tests.nightly>  
</systemProperties>
```

```
$ mvn test -Dtests.nightly=true
```

```
Started J0 PID(26908@MacBook-Pro-4.local).
```

```
Suite: fr.pilato.demo.testframework.RandomTest
```

```
HEARTBEAT J0 PID(26908@MacBook-Pro-4.local): 2016-03-10T18:04:19, stalled for 11.5s at:  
RandomTest.longRunningTest
```

```
HEARTBEAT J0 PID(26908@MacBook-Pro-4.local): 2016-03-10T18:04:29, stalled for 21.5s at:  
RandomTest.longRunningTest
```

```
HEARTBEAT J0 PID(26908@MacBook-Pro-4.local): 2016-03-10T18:04:39, stalled for 31.5s at:  
RandomTest.longRunningTest
```

```
HEARTBEAT J0 PID(26908@MacBook-Pro-4.local): 2016-03-10T18:04:49, stalled for 41.5s at:  
RandomTest.longRunningTest
```

```
OK      50.9s | RandomTest.longRunningTest
```

Sans exagérer toutefois !

timeouts

```
@Test @Nightly @Timeout(millis = 10000)
public void longRunningTest() {
    // ...
}
```

```
$ mvn test -Dtests.nightly=true
```

```
ERROR    10.0s | RandomTest.longRunningTest <<<
> Throwable #1: java.lang.Exception: Test timeout exceeded (>= 10000 msec).
>    at __randomizedtesting.SeedInfo.seed([F4FC818C113EF9C6:A3FC90C16C6643D6]:0)
```




Attention aux zombies !

Pensez à stopper vos Threads

```
@Test
public void stopYourThreads() {
    new Thread(new Runnable() {
        public void run() {
            while (true) {
                try { Thread.sleep(1000L); } catch (InterruptedException e) { } }
        }
    }).start();
}
```

com.carrotsearch.randomizedtesting.ThreadLeakError: 1 thread leaked from SUITE scope at fr.pilato.demo.testframework.RandomTest:

```
1) Thread[id=12, name=Thread-1, state=TIMED_WAITING, group=TGRP-RandomTest]
   at java.lang.Thread.sleep(Native Method)
   at fr.pilato.demo.testframework.RandomTest$1.run(RandomTest.java:180)
   at java.lang.Thread.run(Thread.java:745)
```

```
at __randomizedtesting.SeedInfo.seed([1CD01D6C55CD93C0]:0)
```


Certains zombies sont nos amis

Identifiez-les !

```
@RunWith(RandomizedRunner.class)

public class RandomTest extends RandomizedTest {
    @Test
    public void identifyYourThreads() {
        new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try { Thread.sleep(1000L); } catch (InterruptedException e) { } }
                } }, "friendly-zombie").start();
    }
}
```

Certains zombies sont nos amis

Identifiez-les !

```
@RunWith(RandomizedRunner.class)

public class RandomTest extends RandomizedTest {
    @Test
    public void identifyYourThreads() {
        new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try { Thread.sleep(1000L); } catch (InterruptedException e) { } }
                }, "friendly-zombie").start();
            }
        }
    }

    public class FriendlyZombieFilter implements ThreadFilter {
        public boolean reject(Thread t) {
            if ("friendly-zombie".equals(t.getName())) { return true; }
            return false;
        }
    }
}
```

Certains zombies sont nos amis

Identifiez-les !

```
@RunWith(RandomizedRunner.class)
@ThreadLeakFilters(filters = { FriendlyZombieFilter.class })
public class RandomTest extends RandomizedTest {
    @Test
    public void identifyYourThreads() {
        new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try { Thread.sleep(1000L); } catch (InterruptedException e) { } }
                } }, "friendly-zombie").start();
    }
}

public class FriendlyZombieFilter implements ThreadFilter {
    public boolean reject(Thread t) {
        if ("friendly-zombie".equals(t.getName())) { return true; }
        return false;
    }
}
```

Assert with unknown inputs

Trions, c'est bon pour la planète !

```
@Test
public void checkTestResults() {
    int nbTokens = between(10, 100);
    String[] tokens = new String[nbTokens];
    for (int i = 0; i < nbTokens; i++) {
        tokens[i] = randomAsciiOfLengthBetween(5, 10);
    }
    Arrays.sort(tokens);
}
```

Assert with unknown inputs

Trions, c'est bon pour la planète !

```
@Test
public void checkTestResults() {
    int nbTokens = between(10, 100);
    String[] tokens = new String[nbTokens];
    for (int i = 0; i < nbTokens; i++) {
        tokens[i] = randomAsciiOfLengthBetween(5, 10);
    }
    Arrays.sort(tokens);

    for (int i = 1; i < nbTokens ; i++) {
        assertThat(tokens[i-1], lessThan(tokens[i]));
    }
}
```

Et aussi...



Distributed system

Et si on testait elasticsearch ?



elasticsearch

- Moteur d'indexation, de recherche et d'analytics
- Basé sur Apache Lucene
- Distribué
- Partitionnement
- Réplication
- Scalable horizontalement (et verticalement)

elasticsearch node

- master eligible
 - data
-
- master only : data = false
 - client only : data = false, master = false
 - data only : master = false
 - all (default) : data = true, master = true

elasticsearch client

- use a client from a node
- use a transport client

elasticsearch client

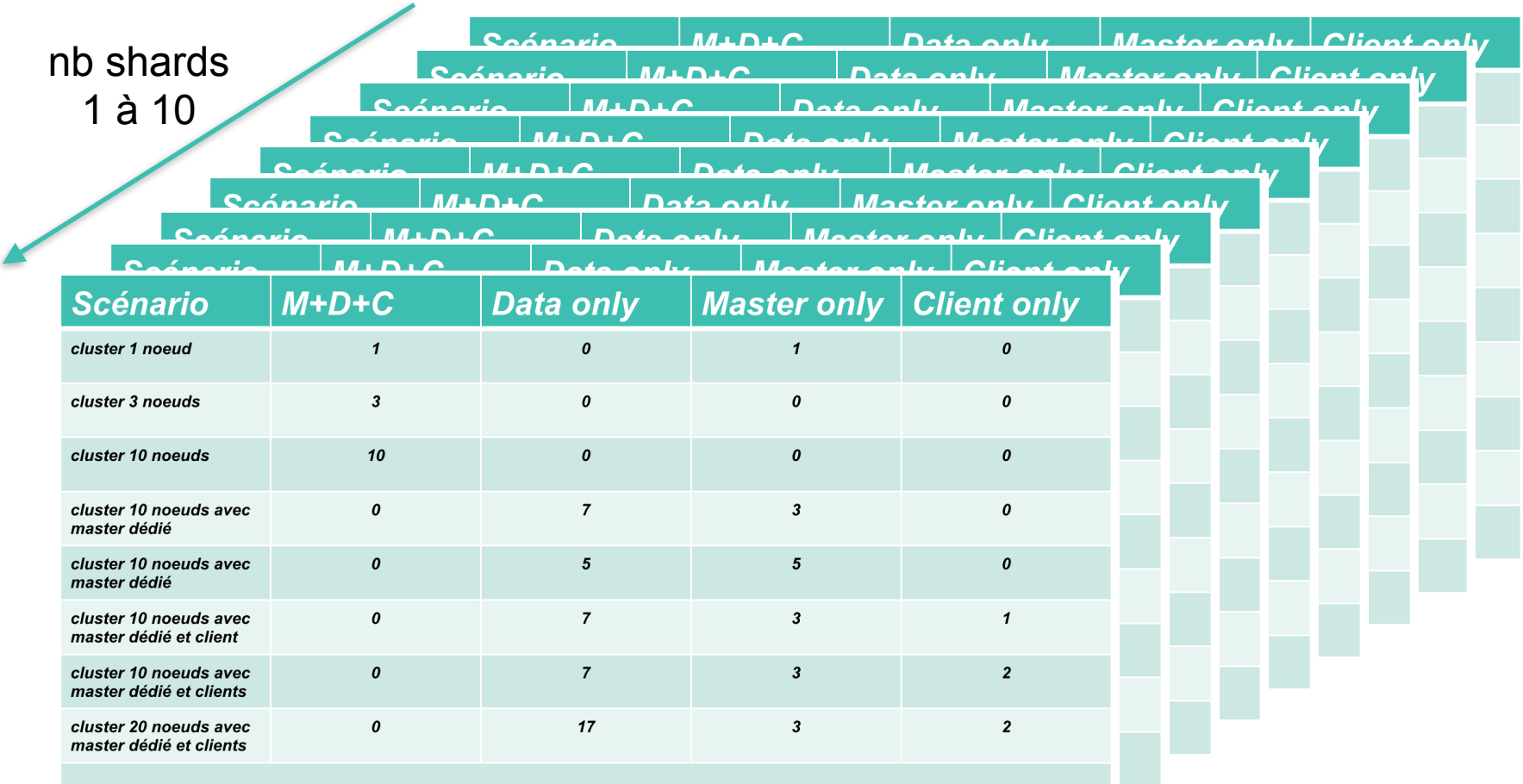
- ~~use a client from a node~~ (deprecated from 5.0)
- use a transport client
- use an http client (coming in 5.0)

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 noeud	1	0	1	0
cluster 3 noeuds	3	0	0	0
cluster 10 noeuds	10	0	0	0
cluster 10 noeuds avec master dédié	0	7	3	0
cluster 10 noeuds avec master dédié	0	5	5	0
cluster 10 noeuds avec master dédié et client	0	7	3	1
cluster 10 noeuds avec master dédié et clients	0	7	3	2
cluster 20 noeuds avec master dédié et clients	0	17	3	2
...				

Mais aussi...

- cluster name
- node name
- index settings
 - nb de shards
 - nb de replicas
 - algorithme de compression
 - ...

nb shards
1 à 10



Scénario	M+D+C	Data only	Master only	Client only
cluster 1 noeud	1	0	1	0
cluster 3 noeuds	3	0	0	0
cluster 10 noeuds	10	0	0	0
cluster 10 noeuds avec master dédié	0	7	3	0
cluster 10 noeuds avec master dédié	0	5	5	0
cluster 10 noeuds avec master dédié et client	0	7	3	1
cluster 10 noeuds avec master dédié et clients	0	7	3	2
cluster 20 noeuds avec master dédié et clients	0	17	3	2
...				

Replicas ?

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 noeud	1	0	1	0
cluster 3 noeuds	3	0	0	0
cluster 10 noeuds	10	0	0	0
cluster 10 noeuds avec master dédié	0	7	3	0
cluster 10 noeuds avec master dédié	0	5	5	0
cluster 10 noeuds avec master dédié et client	0	7	3	1
cluster 10 noeuds avec master dédié et clients	0	7	3	2
cluster 20 noeuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 noeud	1	0	1	0
cluster 3 noeuds	3	0	0	0
cluster 10 noeuds	10	0	0	0
cluster 10 noeuds avec master dédié	0	7	3	0
cluster 10 noeuds avec master dédié	0	5	5	0
cluster 10 noeuds avec master dédié et client	0	7	3	1
cluster 10 noeuds avec master dédié et clients	0	7	3	2
cluster 20 noeuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 noeud	1	0	1	0
cluster 3 noeuds	3	0	0	0
cluster 10 noeuds	10	0	0	0
cluster 10 noeuds avec master dédié	0	7	3	0
cluster 10 noeuds avec master dédié	0	5	5	0
cluster 10 noeuds avec master dédié et client	0	7	3	1
cluster 10 noeuds avec master dédié et clients	0	7	3	2
cluster 20 noeuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 noeud	1	0	1	0
cluster 3 noeuds	3	0	0	0
cluster 10 noeuds	10	0	0	0
cluster 10 noeuds avec master dédié	0	7	3	0
cluster 10 noeuds avec master dédié	0	5	5	0
cluster 10 noeuds avec master dédié et client	0	7	3	1
cluster 10 noeuds avec master dédié et clients	0	7	3	2
cluster 20 noeuds avec master dédié et clients	0	17	3	2

Compression ?

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0
cluster 10 nœuds	10	0	0	0
cluster 10 nœuds avec master dédié	0	7	3	0
cluster 10 nœuds avec master dédié	0	5	5	0
cluster 10 nœuds avec master dédié et client	0	7	3	1
cluster 10 nœuds avec master dédié et clients	0	7	3	2
cluster 20 nœuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0
cluster 10 nœuds avec master dédié	0	7	3	0
cluster 10 nœuds avec master dédié	0	5	5	0
cluster 10 nœuds avec master dédié et client	0	7	3	1
cluster 10 nœuds avec master dédié et clients	0	7	3	2
cluster 20 nœuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0
cluster 10 nœuds	10	0	0	0
cluster 10 nœuds avec master dédié	0	7	3	0
cluster 10 nœuds avec master dédié	0	5	5	0
cluster 10 nœuds avec master dédié et client	0	7	3	1
cluster 10 nœuds avec master dédié et clients	0	7	3	2
cluster 20 nœuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0
cluster 10 nœuds	10	0	0	0
cluster 10 nœuds avec master dédié	0	7	3	0
cluster 10 nœuds avec master dédié	0	5	5	0
cluster 10 nœuds avec master dédié et client	0	7	3	1
cluster 10 nœuds avec master dédié et clients	0	7	3	2
cluster 20 nœuds avec master dédié et clients	0	17	3	2

nb shards
1 à 10

Scénario	M+D+C	Data only	Master only	Client only
cluster 1 nœuds	1	0	1	0
cluster 2 nœuds	3	0	0	0
cluster 10 nœuds	10	0	0	0
cluster 10 nœuds avec master dédié	0	7	3	0
cluster 10 nœuds avec master dédié	0	5	5	0
cluster 10 nœuds avec master dédié et client	0	7	3	1
cluster 10 nœuds avec master dédié et clients	0	7	3	2
cluster 20 nœuds avec master dédié et clients	0	17	3	2

... ?







Créer un cluster

au hasard

```
return new InternalTestCluster(nodeMode, seed, createTempDir(),
    minNumDataNodes, maxNumDataNodes,
    InternalTestCluster.clusterName(scope.name(), seed) + "-cluster",
    nodeConfigurationSource, getNumClientNodes(),
    InternalTestCluster.DEFAULT_ENABLE_HTTP_PIPELINE, nodePrefix,
    enableMockModules);

// ... Compute cluster settings based on ^^^

logger.info("Setup cluster [{}] using [{}] data nodes and [{}] client nodes",
    clusterName, numSharedDataNodes, numSharedClientNodes);
```

Créer un index

au hasard aussi

```
protected int minimumNumberOfShards() { return DEFAULT_MIN_NUM_SHARDS; }
protected int maximumNumberOfShards() { return DEFAULT_MAX_NUM_SHARDS; }
protected int minimumNumberOfReplicas() { return 0; }
protected int maximumNumberOfReplicas() {
    int maxNumReplicas = Math.max(0, numDataNodes() - 1);
    return frequently() ? Math.min(1, maxNumReplicas) : maxNumReplicas;
}
protected int numberOfShards() {
    return between(minimumNumberOfShards(), maximumNumberOfShards());
}
protected int numberOfReplicas() {
    return between(minimumNumberOfReplicas(), maximumNumberOfReplicas());
}

builder.put(SETTING_NUMBER_OF_SHARDS, numberOfShards())
        .put(SETTING_NUMBER_OF_REPLICAS, numberOfReplicas());
```

Assigne tous les settings possibles

toujours au hasard

```
if (random.nextBoolean()) { builder.put(AUTO_THROTTLE, false); }
if (random.nextBoolean()) { builder.put(INDEX_CACHE_REQUEST_ENABLED,
    random.nextBoolean()); }
if (random.nextBoolean()) { builder.put("index.shard.check_on_startup",
    randomFrom(random, "false", "checksum", "true")); }
if (random.nextBoolean()) { builder.put(INDEX_TRANSLOG_DISABLE_FLUSH,
    random.nextBoolean()); }
if (random.nextBoolean()) { builder.put(INDEX_TRANSLOG_FLUSH_THRESHOLD_OPS,
    randomIntBetween(random, 1, 10000)); }
if (random.nextBoolean()) { builder.put(INDEX_TRANSLOG_DURABILITY,
    randomFrom(random, Translog.Durability.values())); }
if (random.nextBoolean()) {
    builder.put(INDEX_TRANSLOG_FS_TYPE,
        randomFrom(random, TranslogWriter.Type.values()));
    if (rarely(random)) { builder.put(INDEX_TRANSLOG_SYNC_INTERVAL, 0); }
    else { builder.put(INDEX_TRANSLOG_SYNC_INTERVAL,
        randomIntBetween(random, 100, 5000), TimeUnit.MILLISECONDS);
    }
}
```


*If you build it
he will come.*



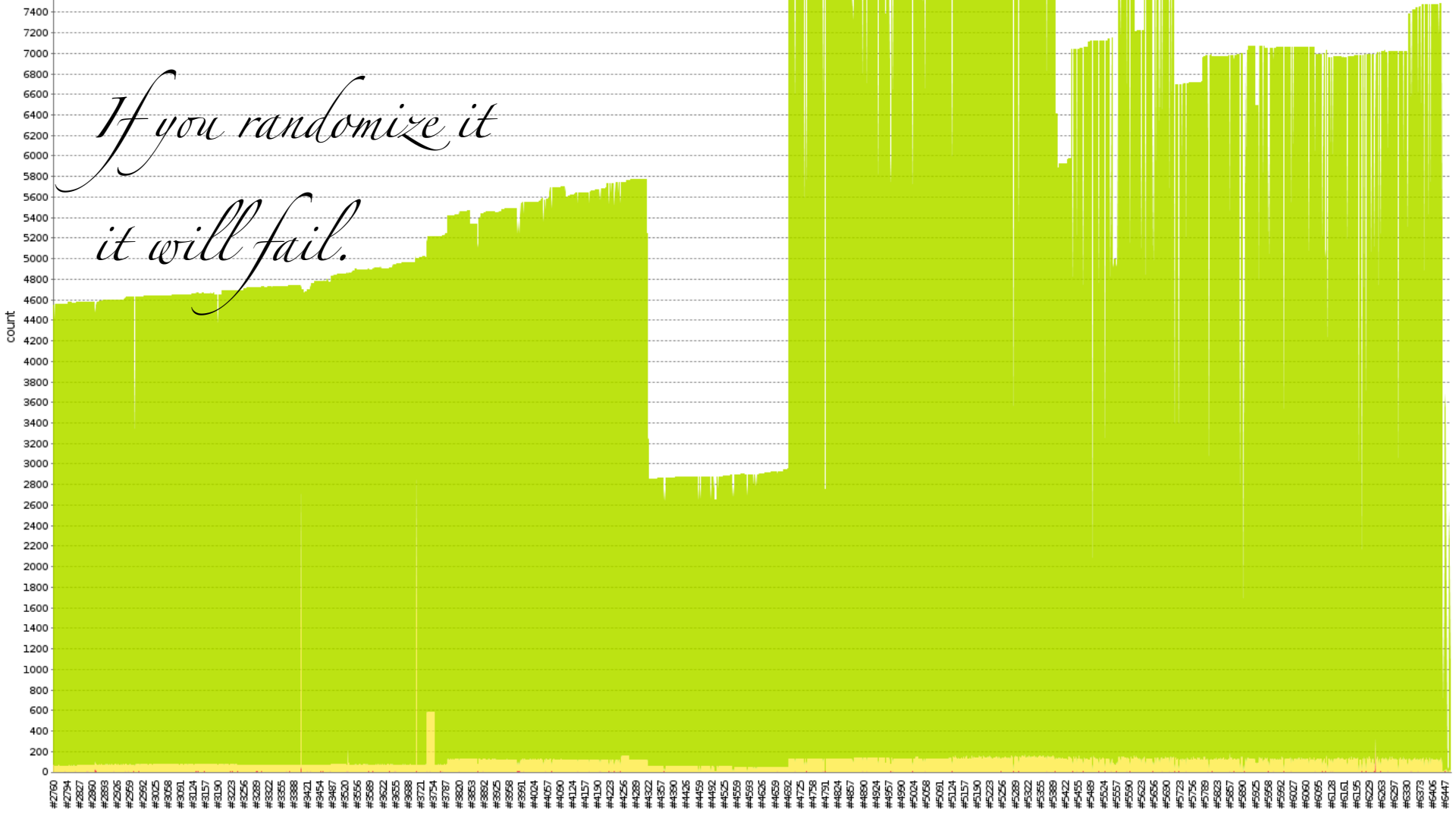


Plate-forme aléatoire

Paramètres de JVM aléatoires

- ES Server Master on Centos
- ES Server Master on Debian
- ES Server Master on Medium
- ES Server Master on Metal
- ES Server Master on Oracle
- ES Server Master on Oracle 6
- ES Server Master on Oracle 7
- ES Server Master on Small
- ES Server Master on Strong
- ES Server Master on Suse
- ES Server Master on Window 2012
- ES Server Master on Windows 2008





David Pilato

Developer | Evangelist

@dadoonet

Le hasard fait bien les tests

Thanks !



Breizh C@mp
La conférence à l'Ouest