Improving Performance Analysis of Software System Versions Using Change-Based Test Selection

David Georg Reichelt ¹  Fabian Scheller ²

¹Abteilung Betriebliche Informationssysteme, Universität Leipzig

²Institut für Infrastruktur und Ressourcenmanagement, Universität Leipzig

6. November 2015
1. Performance Analysis of Software System Versions (PeASS)

2. Process

3. Related Work

4. Summary and Future Work
Basic Idea

- Research question: which performance problem classes exist on code level?
- State of the art: little systematic research on performance problem classes on code level
- Approach: analyse performance of units tests of versions of a program
  - Detect changes
  - Detect problems
  - Classify problems
Assumptions

- basic assumptions
  - performance changes during software development
  - performance of unit tests corresponds to performance of program

  ⇒ relevant program classes: frameworks or backend components
Example: Apache Commons IO

- performance change between 560660 and 584162
- relevant commits
  - 561233 - Fixing svn locations after TLP move
  - 561417 - Move Commons TLP changes
  - 561491 - Apache Apache
  - 561555 - Apache Apache
  - 561564 - finish long-overdue-to-be-finished removal of documented lang dependency
  - 561628 - Commons TLP: directory [...] has moved to "dist/commons"[..]
  - 567499 - Updated commons parent version to 4.
  - 568574 - Removing the 'Commons ' from the names [..]
  - 568588 - Fixing the user svn url
  - 568979 - Changing name to 'Commons Xxx'
  - 584162 - IO-126 Add facility to specify case sensitivity for prefix and suffix file filter
Example: Measurement
Example: Diff

```java
/// >>>/projekte/commons-10/src/test/java/org/apache/commons/io/output/LockableFileWriterTest.java
++++ /// >>>/projekte/commons-10/src/test/java/org/apache/commons/io/output/LockableFileWriterTest.java
@@    -19,7 +19,6 @@
101   import java.io.File;
102   import java.io.IOException;
103   import java.io.Writer;
104   -import java.nio.charset.UnsupportedCharsetException;
105
106   import org.apache.commons.io.IOSUtilities;
107   import org.apache.commons.io.testtools.FileBasedTestCase;
108   @@ -160,12 +159,12 @@
109   }
110
111   //-----------------------------------------------
112   - public void testConstructor_File_encoding_badEncoding() throws IOException {
113   + public void testConstructor_File_encoding_badEncoding() {
114       Writer writer = null;
115       try {
116           writer = new LockableFileWriter(file, "BAD-ENCODE");
117           fail();
118       - } catch (UnsupportedCharsetException ex) {
119       + } catch (IOException ex) {
120           // expected
121           assertFalse(file.exists());
122           assertFalse(lockFile.exists());
```
Steps

- main problem: performance measurements are instable
  - background processes
  - just-in-time compilation
  - ..
- approach: measurement and refined measurement
  - measurement of performance for all testcases in all versions
  - identification of performance changes
  - identification of performance problems
Performance Measurement

Process:
1. Version Download
2. Performance Test Generation
3. Process-instrumentation
4. Test Execution
5. Version Control System
6. Testframework
7. Build Tool

Flow:
- [Yes] from Version Download to Performance Test Generation
- [No] from Version Download to Next Version?
- [Yes] from Process-instrumentation to Test Execution
- [No] from Test Execution to Version Download

Next Version? [Yes] or [No]
Identification of Performance Changes

Change Candidate Identification

Boundary Value, Width

Test

Remeasurement

Iterations

Change Identification

Boundary Value
Enhanced PeASS process

- problems of old process
  - many unnecessary measurements
  - heuristics lead to lost performance changes

- enhanced process
  - change-based test selection
    - usage of kieker instrumentation and trace analysis
    - selecting relevant tests for re-run
  - measurement for selected unit tests
  - identification of performance problems
Change-based Test Selection

(I) Initial Dependency Construction

(II) VCS Diff Analysis

(III) Marked Test Saving

(IV) Continuous Dependency Construction

More Versions?

[no] [yes]
Enhanced PeASS process

- find first performance change in Apache Commons IO with 4000 iterations each
  - reduced test executions: 428 tests, 8 h 4 min
  - all test executions: 6320 tests, ∼ 119 h
- counts of normal and reduced tests for chosen Apache Commons projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Normal Tests</th>
<th>Reduced Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commons BCEL</td>
<td>100521</td>
<td>12109</td>
</tr>
<tr>
<td>Commons BeanUtils</td>
<td>97178</td>
<td>8261</td>
</tr>
<tr>
<td>Commons Collections</td>
<td>44058</td>
<td>842</td>
</tr>
<tr>
<td>Commons IO</td>
<td>659190</td>
<td>77259</td>
</tr>
<tr>
<td>Commons Jelly</td>
<td>298</td>
<td>13</td>
</tr>
</tbody>
</table>
Related Work

- green mining (Hindle et al., 2014)
- search of performance errors in bug tracker (Jin et al., 2012, Nistor et al., 2013)
- application of performance tests to repositories (Horký et al., 2013) (Heger et al., 2013)
- (performance antipatterns on architecture level (Smith et al., 2003))
Summary

- basic idea: examine development of performance of unit tests during software development
- goal: classification of typical performance problems
- method: change-based test selection
  - reconstruct call hierarchy with kieker
  - analyse VCS diffs
  - only re-run tests which are influenced by change
Future Work

- reusage of change-based test selection
  - usage for load tests
  - usage for performance unit tests

- current work for statistic rigor of PeASS
  - measurement setup: vm runs, warmup, measurement executions
  - statistical evaluation: confidence intervals, boundary values

- implementing root-cause isolation of performance problems
Thanks for your attention!
Any questions?

For further questions:

David Georg Reichelt
Abteilung Betriebliche Informationsysteme
Universität Leipzig
reichelt@informatik.uni-leipzig.de
Bibliography


