Identifying Performance Changes Using Peass
FOSDEM 2021, Continuous Integration and Continuous Deployment

David Georg Reichelt\textsuperscript{1}

\textsuperscript{1}Universität Leipzig, University Computing Centre, Research and Development

7th February 2021

FKZ 01IS20032D
Eliminated a lot of checkstyle warnings.

git-svn-id: https://svn.apache.org/repos/asf/jakarta/commons/proper/fileupload/trunk@479484 13f79535-47bb-0316-9956-ffa459ede88

master commons-fileupload-1.4 FILEUPLOAD_1_3_RC1

jochenw committed on 27 Nov 2006
1 parent fdf011a commit 4ed6e923cb2033272fcb993978d59e325009a5aa

Showing 15 changed files with 751 additions and 317 deletions.
What is faster?

```java
StringBuilder buf = new StringBuilder(16);
buf.append("Hello\World");
return buf.toString();
```
What is faster?

```java
StringBuilder buf = new StringBuilder(16);
buf.append("Hello\u2423World");
return buf.toString();
```
What is faster?

```java
StringBuilder buf = new StringBuilder(16);
buf.append("Hello\u0002World");
return buf.toString();
```

Method: Unit Test Assumption

Software
- Component 1
- Component 2
- Component 3

Load Test
- JMeter
- IBM

Benchmark
- Hyperledger
- Caliper
- JMH
Method: Unit Test Assumption

Software

Component 1  Component 2  Component 3

Load Test

Benchmark

0.4% of all projects

Software

Component 1  Component 2  Component 3

Load Test

Benchmark
Method: Unit Test Assumption

Load Test

Benchmark

Software

Component 1

Component 2

Component 3

Unit Test 1

Unit Test 2

Unit Test 3

0.4% of all projects

33.2% of all projects
Method: Unit Test Assumption

Unit-Test-Assumption:
The Performance of relevant use cases of a program correlates with the performance of at least a part of its unit tests.

- **Load Test**: JMeter
- **Benchmark**: Hyperledger Caliper JMH

Software Components:
- Component 1
  - Unit Test 1: 0.4% of all projects
- Component 2
  - Unit Test 2
- Component 3
  - Unit Test 3: 33.2% of all projects
Approach of Peass

Repository

Version 1
Test 1  Test 2

Version 2
Test 1  Test 2
Approach of Peass

Performance Measurement

- VM Start 1
- VM Warmup Iterations
- Measurement Iterations
Approach of Peass

Performance Measurement

- VM Start 1 → VM Warmup Iterations → Measurement Iterations
- VM Start 2 → VM Warmup Iterations → Measurement Iterations
- VM Start 3 → VM Warmup Iterations → Measurement Iterations
- VM Start 4 → VM Warmup Iterations → Measurement Iterations
- VM Start 5 → VM Warmup Iterations → Measurement Iterations
- VM Start 6 → VM Warmup Iterations → Measurement Iterations
- VM Start 7 → VM Warmup Iterations → Measurement Iterations
- VM Start 8 → VM Warmup Iterations → Measurement Iterations
- VM Start 9 → VM Warmup Iterations → Measurement Iterations
Approach of Peass

Repository

- **Version 1**
  - Test 1
  - Test 2

- **Version 2**
  - Test 1
  - Test 2

Measurement

- **Version 2**
  - Test 1
  - Test 2

![Diagram showing repository and measurement process]
Approach of Peass

Regression Test Selection

Test 1  Test 2  Test 3  Test 4  Test 5
Approach of Peass

Regression Test Selection

Repository

Test 1

Class 1

Change

Test 2

Class 2

Test 3

Class 3

Test 4

Class 4

Test 5

Class 5
Approach of Peass

Repository

Version 1
Test 1  Test 2

Version 2
Test 1  Test 2

Regression Test Selection

Version 1
Test 1  Test 2

Version 2
Test 1  Test 2

Measurement

Version 2
Test 1  Test 2

This diagram illustrates the process of selecting regression tests for different versions of a software repository.
Approach of Peass

Repository

Version 1
- Test 1
- Test 2

Version 2
- Test 1
- Test 2

Regression Test Selection

Version 1
- Test 1
- Test 2

Version 2
- Test 1
- Test 2

Root-Cause Analysis

Measurement

Version 2
- Test 1
- Test 2
Demo

Demo
Next Steps

- reliable and fast...
  - measurement: speedup with parallelization, cgroups, ...
  - root cause analysis: call tree node selection, measurement probe optimization, ...

- practical use
  - use for open source projects
  - use for projects of partners

- get involved
  - try it and tell us your experience
  - open for PRs
Thanks for your attention!

David Georg Reichelt
Universitätsrechenzentrum
Universität Leipzig
david_georg.reichelt@uni-leipzig.de