

Identifying Performance Changes Using Peass

FOSDEM 2021, Continuous Integration and Continuous Deployment

David Georg Reichelt¹

¹Universität Leipzig, University Computing Centre, Research and Development

7th February 2021

FKZ 01IS20032D

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

<> Code

Pull requests 2

Actions

Projects

Security


...

Eliminated a lot of checkstyle warnings.

[Browse files](#)

git-svn-id: <https://svn.apache.org/repos/asf/jakarta/commons/proper/fileupload/trunk@479484> 13f79535-47bb-0310-9956-ffa450edef68

master commons-fileupload-1.4 FILEUPLOAD_1_3_RC1

 jochenw committed on 27 Nov 2006

1 parent fdf011a commit 4ed6e923cb2033272fcb993978d69e325990a5aa

Showing 15 changed files with 751 additions and 317 deletions.

Unified

Split

src/checkstyle/fileupload_checks.xml	+4 -1	■■■■■
src/java/org/apache/commons/fileupload/DiskFileUpload.java	+1 -0	■□□□□
src/java/org/apache/commons/fileupload/FileItemIterator.java	+15 -7	■■■■■
src/java/org/apache/commons/fileupload/FileItemStream.java	+11 -5	■■■■■
src/java/org/apache/commons/fileupload/FileUpload.java	+4 -4	■■■■■
src/java/org/apache/commons/fileupload/FileUploadBase.java	+312 -135	■■■■■
src/java/org/apache/commons/fileupload/FileUploadException.java	+38 -16	■■■■■
src/java/org/apache/commons/fileupload/MultipartStream.java	+158 -69	■■■■■

What is faster?

```
1  StringBuilder buf
2    = new StringBuilder(16);
3  buf.append("Hello_World");
4
5
6  return buf.toString();
```

```
1  StringBuilder buf
2    = new StringBuilder(16);
3  buf.append("Hello")
4    .append("_")
5    .append("World");
6  return buf.toString();
```

What is faster?

```
1  StringBuilder buf
2    = new StringBuilder(16);
3  buf.append("Hello_World");
4
5
6  return buf.toString();
```

```
1  StringBuilder buf
2    = new StringBuilder(16);
3  buf.append("Hello")
4    .append("_")
5    .append("World");
6  return buf.toString();
```

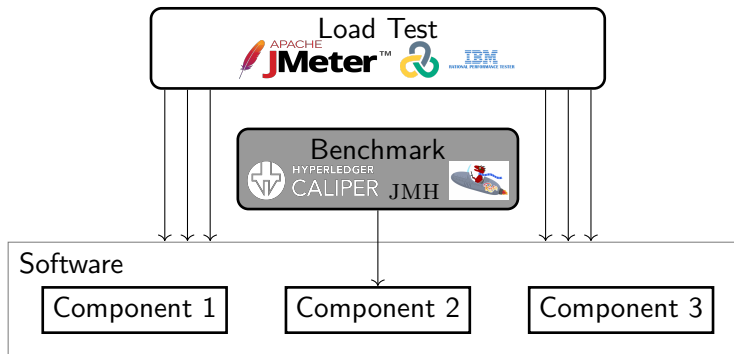
What is faster?

```
1  StringBuilder buf
2    = new StringBuilder(16);
3  buf.append("Hello_World");
4
5
6  return buf.toString();
```

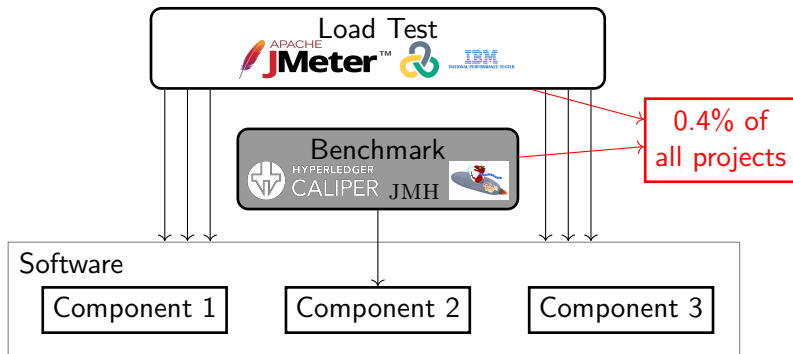
```
1  StringBuilder buf
2    = new StringBuilder(16);
3  buf.append("Hello")
4    .append("_")
5    .append("World");
6  return buf.toString();
```

David Georg Reichelt, Stefan Kühne, Wilhelm Hasselbring: On the Validity of Performance Antipatterns at Code Level, Symposium on Software Performance, 2019.

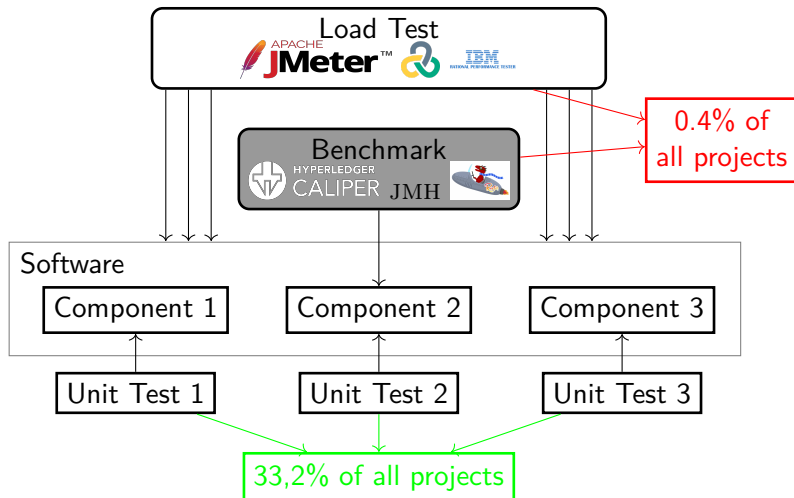
Method: Unit Test Assumption



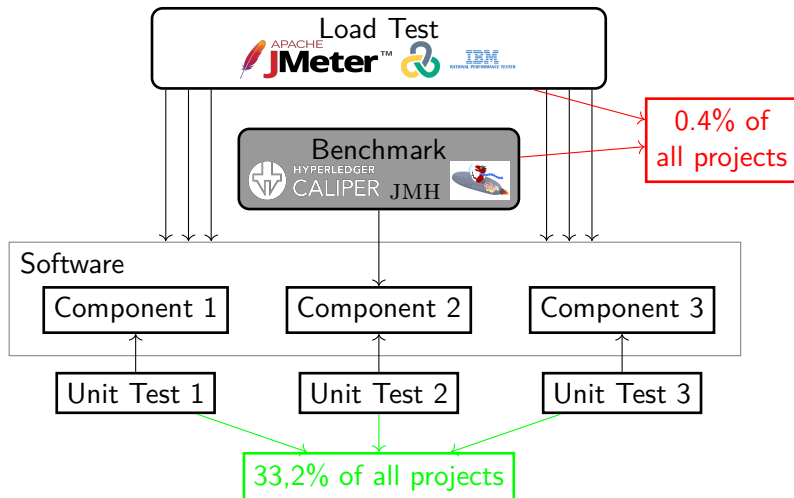
Method: Unit Test Assumption



Method: Unit Test Assumption



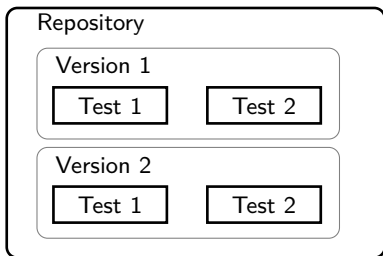
Method: Unit Test Assumption



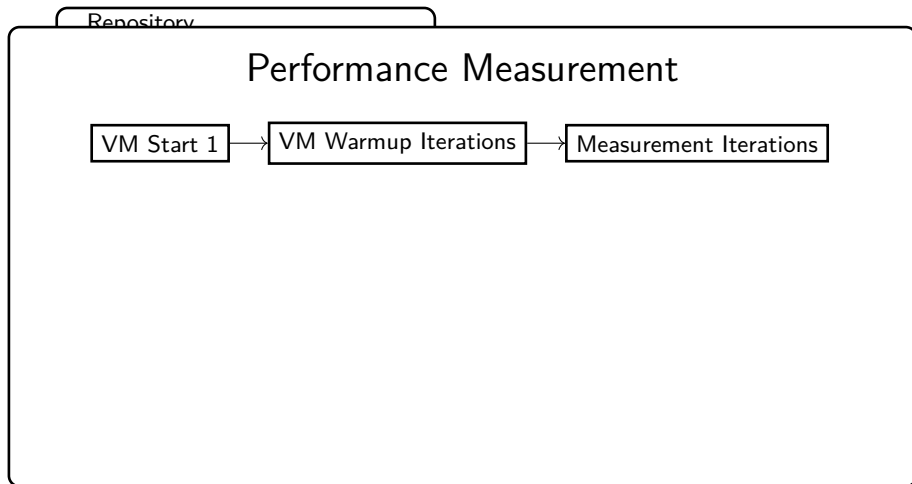
Unit-Test-
Assumption:

The Performance of
relevant use cases of
a program correlates
with the performance
of at least **a part** of
its unit tests

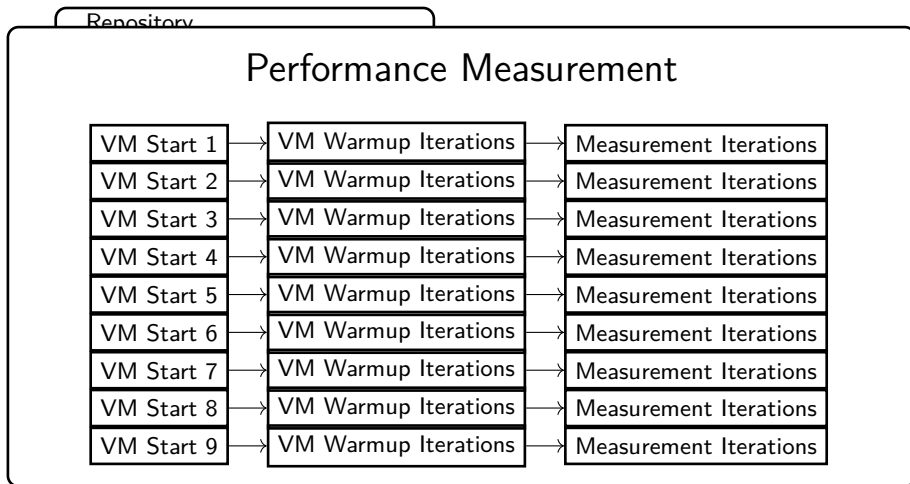
Approach of Peass



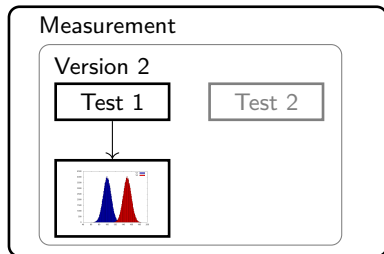
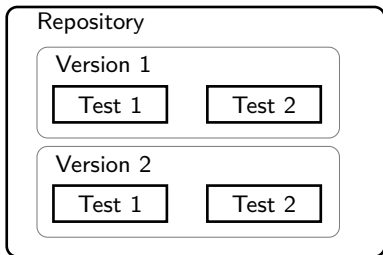
Approach of Peass



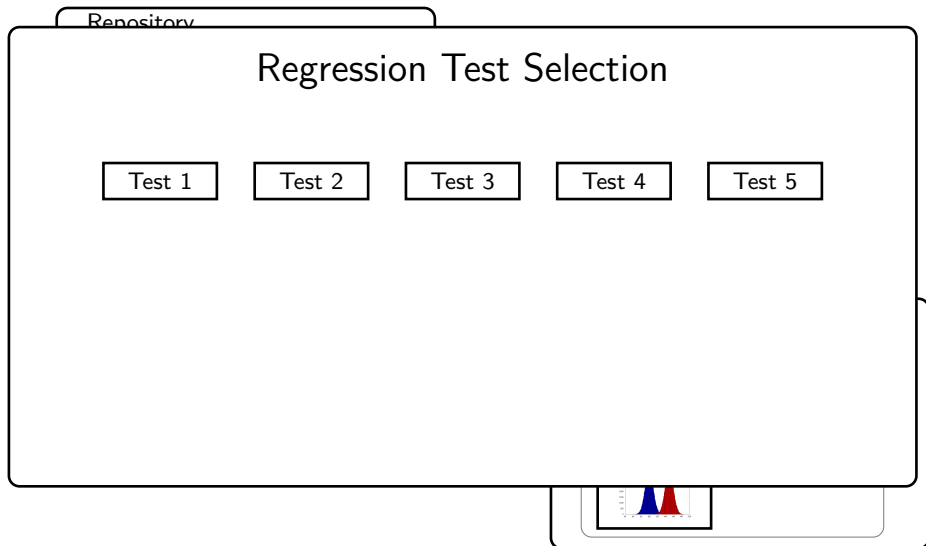
Approach of Peass



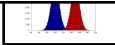
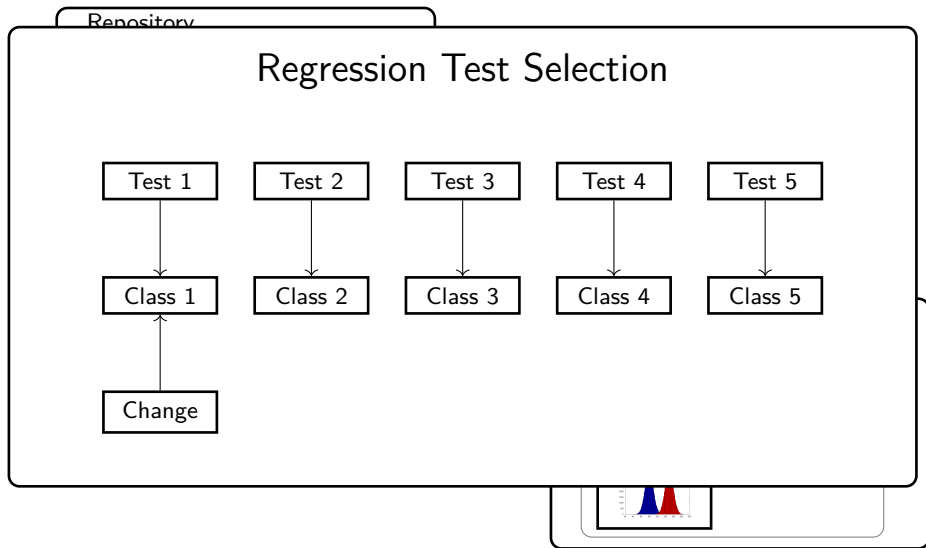
Approach of Peass



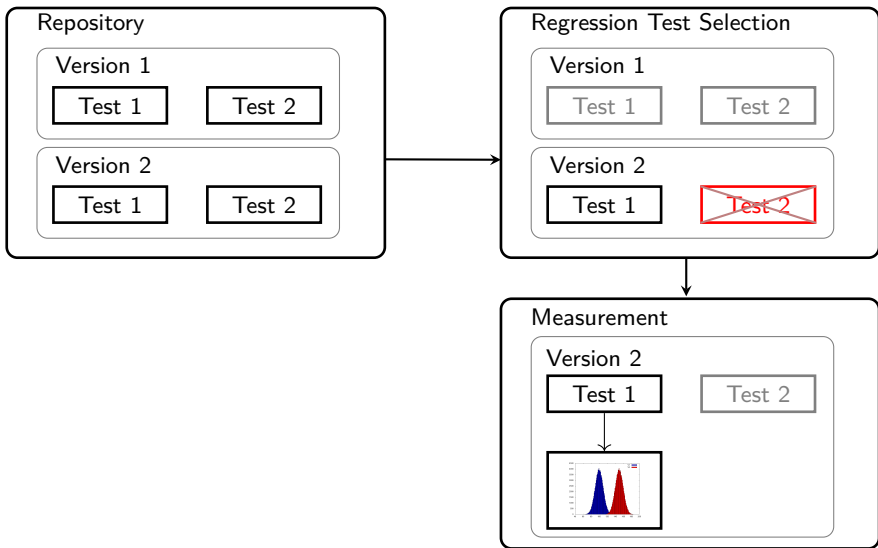
Approach of Peass



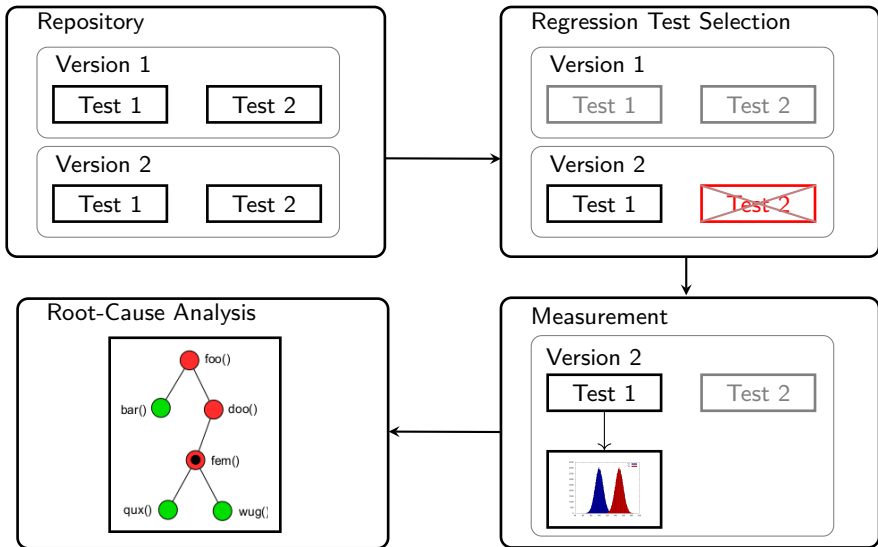
Approach of Peass



Approach of Peass



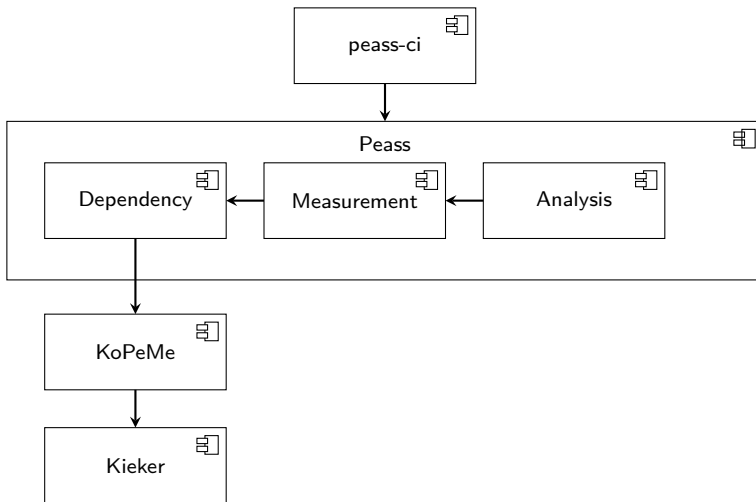
Approach of Peass



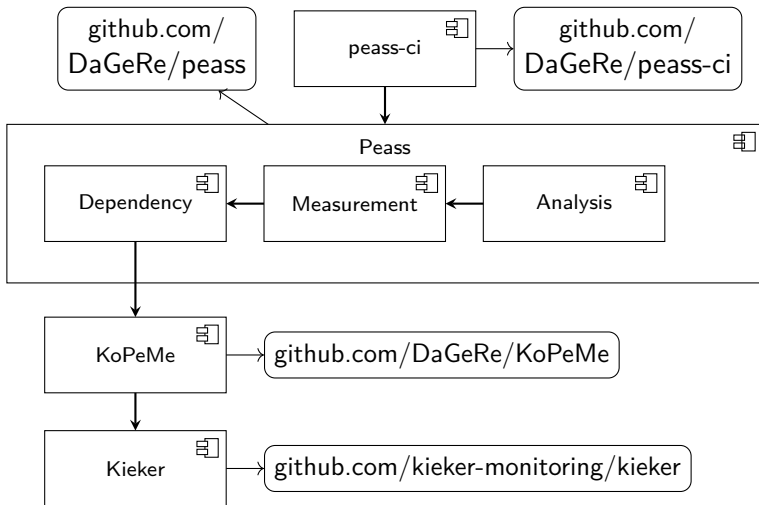
Demo

Demo

Components



Components



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

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung