BUMP: A Benchmark of Reproducible Breaking Dependency Updates

Frank Reyes, Yogya Gamage, Gabriel Skoglund, Benoit Baudry, Martin Monperrus

KTH Royal Institute of Technology

SANER 2024

Outline

1. Introduction
2. What is a Breaking dependency update?
3. BUMP
4. Statistics
5. Conclusions

Introduction

pom.xml

```
<dependencies>
   <dependency>
      <groupId>org.A</groupId>
       <artifactId>abc</artifactId>
       <version>1.318</version>
   </dependency>
   <dependency>
      <groupId>org.B</groupId>
      <artifactId>ghj</artifactId>
      <version>4.12.0</version>
   </dependency>
</dependencies>
```

Introduction

pom.xml

```
@@ -86,7 +86,7 @@
86  86          <dependency>
87  87              <groupId>org.hibernate.validator</groupId>
88  88              <artifactId>hibernate-validator</artifactId>
89      -           <version>6.1.7.Final</version>
    89  +           <version>7.0.3.Final</version>
90  90          </dependency>
91  91          <dependency>
92  92              <groupId>jakarta.el</groupId>
```

Dependencies

Update

Updating dependencies is a risky task that can break existing functionality

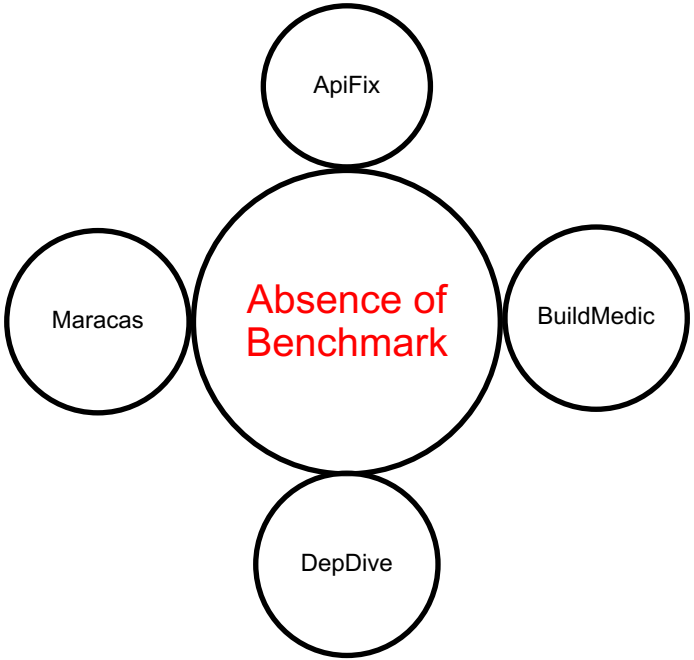
Breaking dependency updates

Motivation

Causes of breaking dependency updates

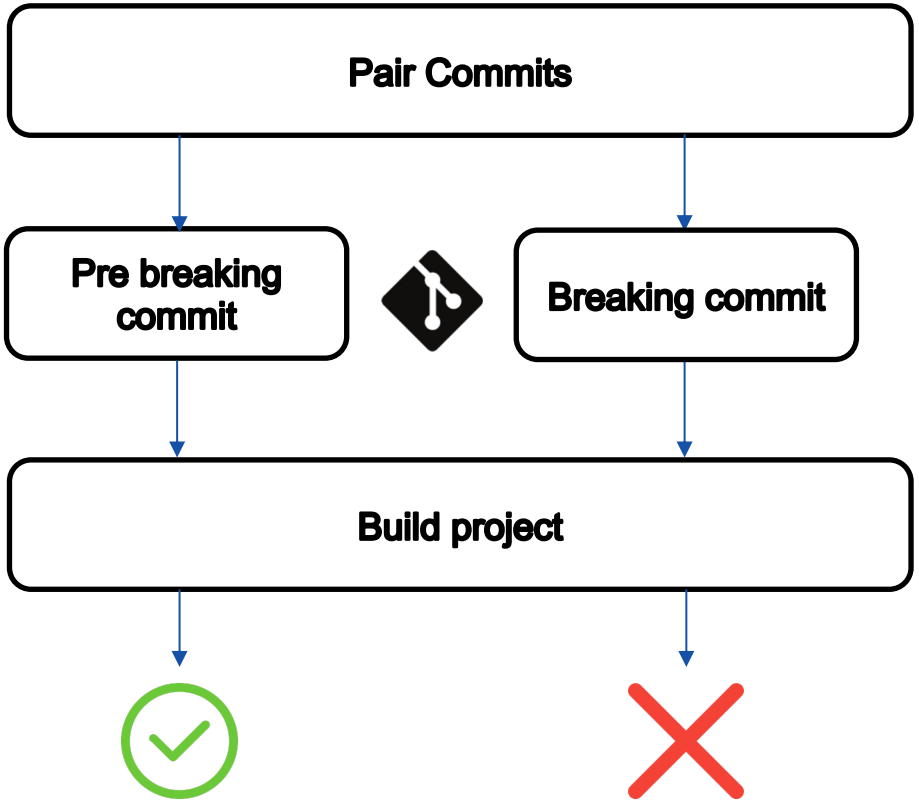
- changes in the dependency interface (API)
- changes in the behaviour of the updated dependencies

API Repair Tools



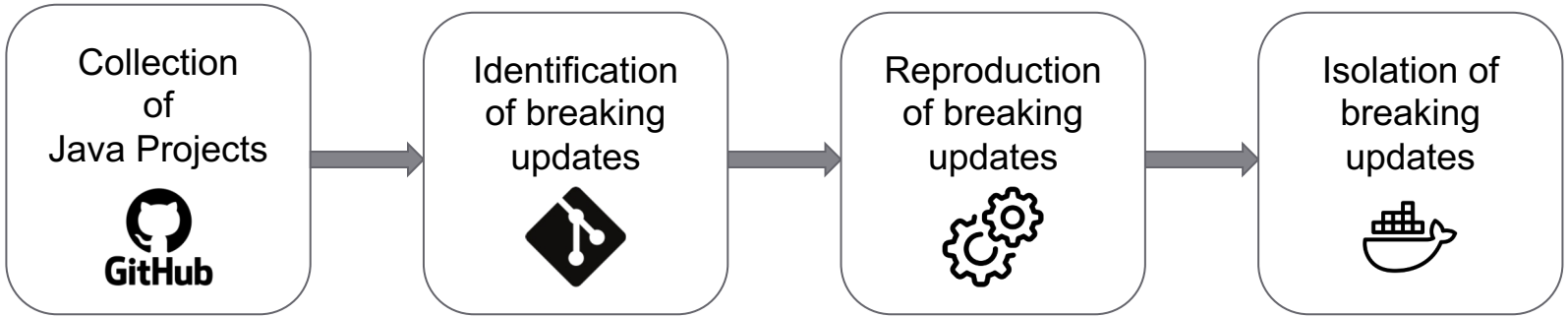
C. Macho, S. McIntosh, and M. Pinzger, “Automatically repairing dependency-related build breakage,” SANER 2018.

What is a Breaking dependency update?



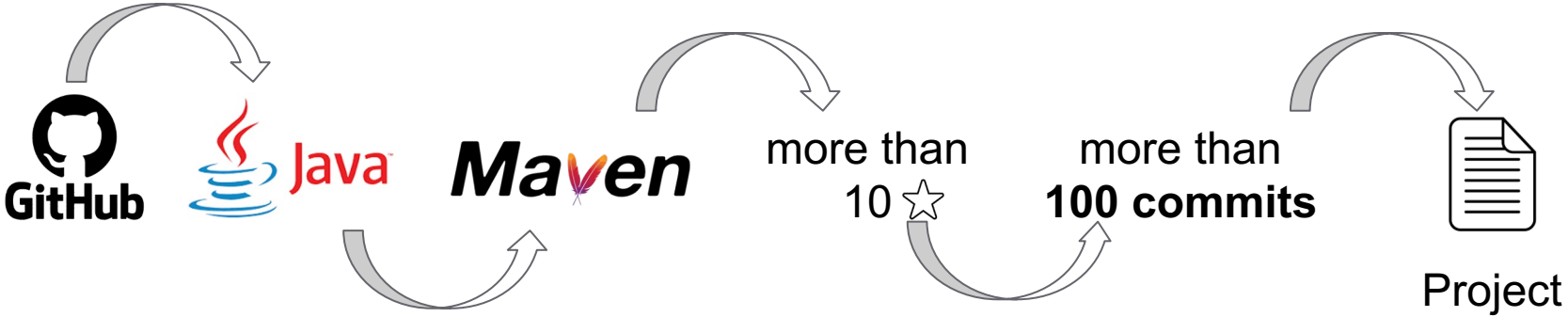

```
<dependency>
        <groupId>net.sf.jasperreports</groupId>
        <artifactId>jasperreports</artifactId>
        <version>6.18.1</version>
        <version>6.19.1</version>
</dependency>
</dependencies>
```

Building the BUMP Benchmark



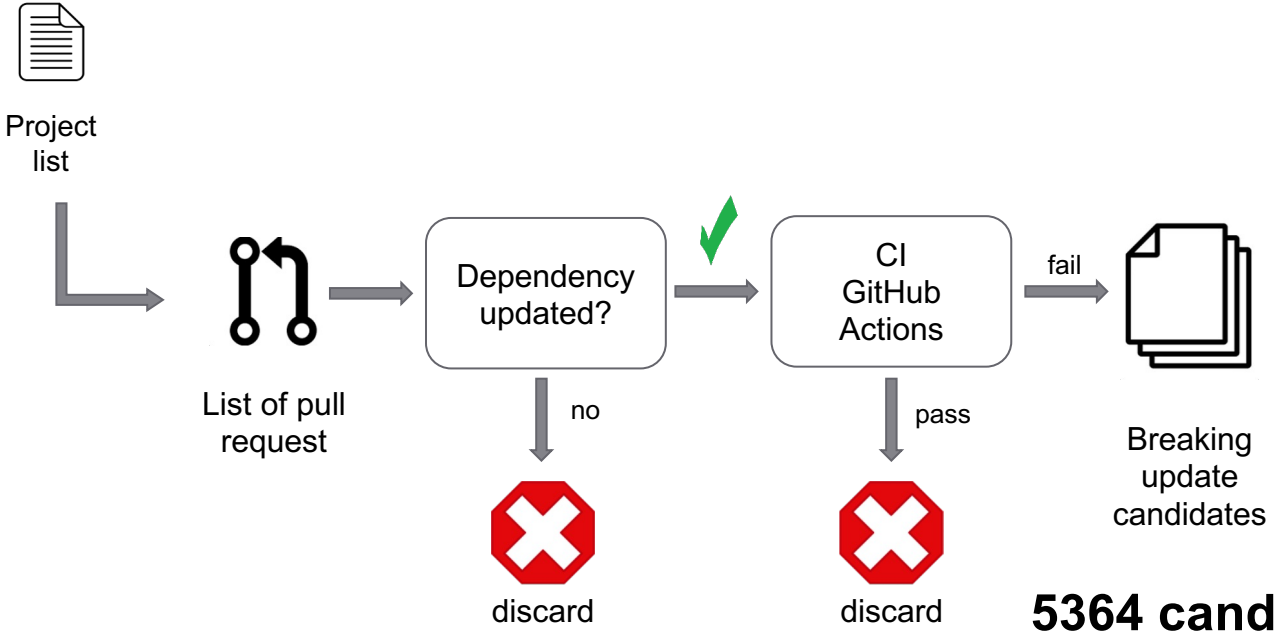
https://github.com/chains-project/bump

Collection of Java projects



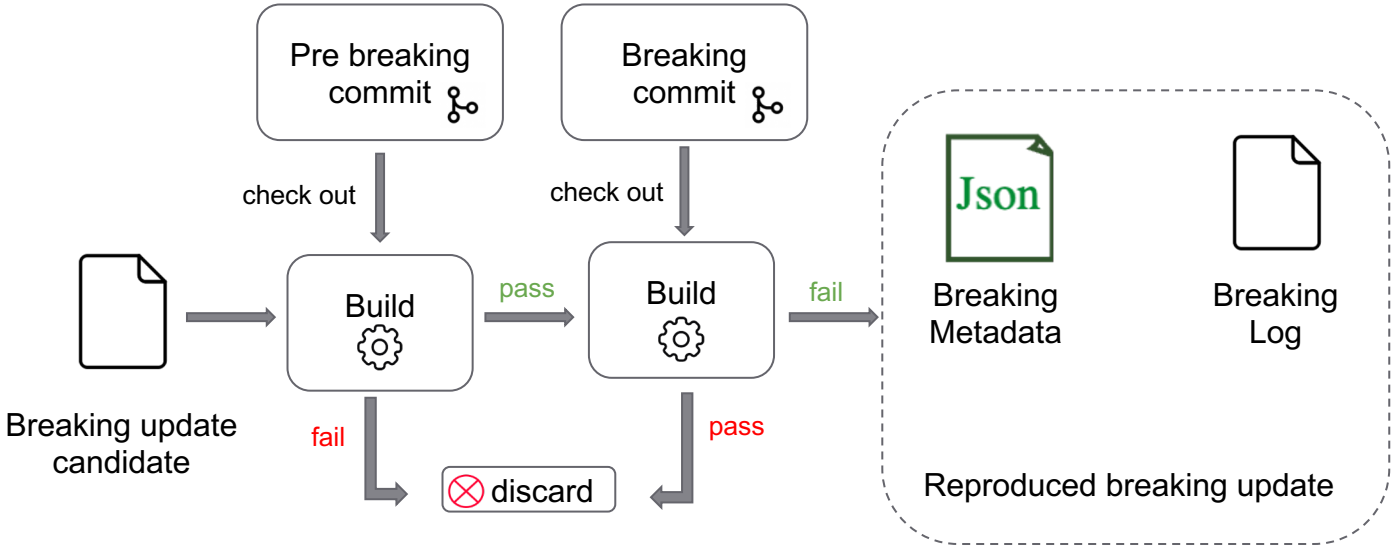
4148 projects

Identification of breaking updates



5364 candidates

Reproduction of breaking updates



Reproduction of breaking updates

- **628** breaking dependency updates

- Flaky tests
- Environmental dependence
- Exotic failures

- **571** breaking dependency updates


```
{
  "url":
↪   "https://github.com/xdev-software/biapi/pull/69",
  "project": "biapi",
  "projectOrganisation": "xdev-software",
  "breakingCommit":
↪   "0abf7148300f40a1da0538ab060552bca4a2f1d8",
  "prAuthor": "bot",
  "preCommitAuthor": "human",
  "breakingCommitAuthor": "bot",
  "updatedDependency": {
    "dependencyGroupID": "net.sf.jasperreports",
    "dependencyArtifactID": "jasperreports",
    "previousVersion": "6.18.1",
    "newVersion": "6.19.1",
    "dependencyScope": "compile",
    "versionUpdateType": "minor",
    ...
    "updatedFileType": "JAR"
  },
  "preCommitReproductionCommand": "docker run
↪   ghcr.io/chains-project/breaking-updates:0ab⌋
↪   f7148300f40a1da0538ab060552bca4a2f1d8-pre",
  "breakingUpdateReproductionCommand": "docker run
↪   ghcr.io/chains-project/breaking-updates:0ab⌋
↪   f7148300f40a1da0538ab060552bca4a2f1d8-breaking",
  "javaVersionUsedForReproduction": "11",
  "failureCategory": "COMPILATION_FAILURE"
}
```


author's information

dependency information

isolation information

Failure Categories

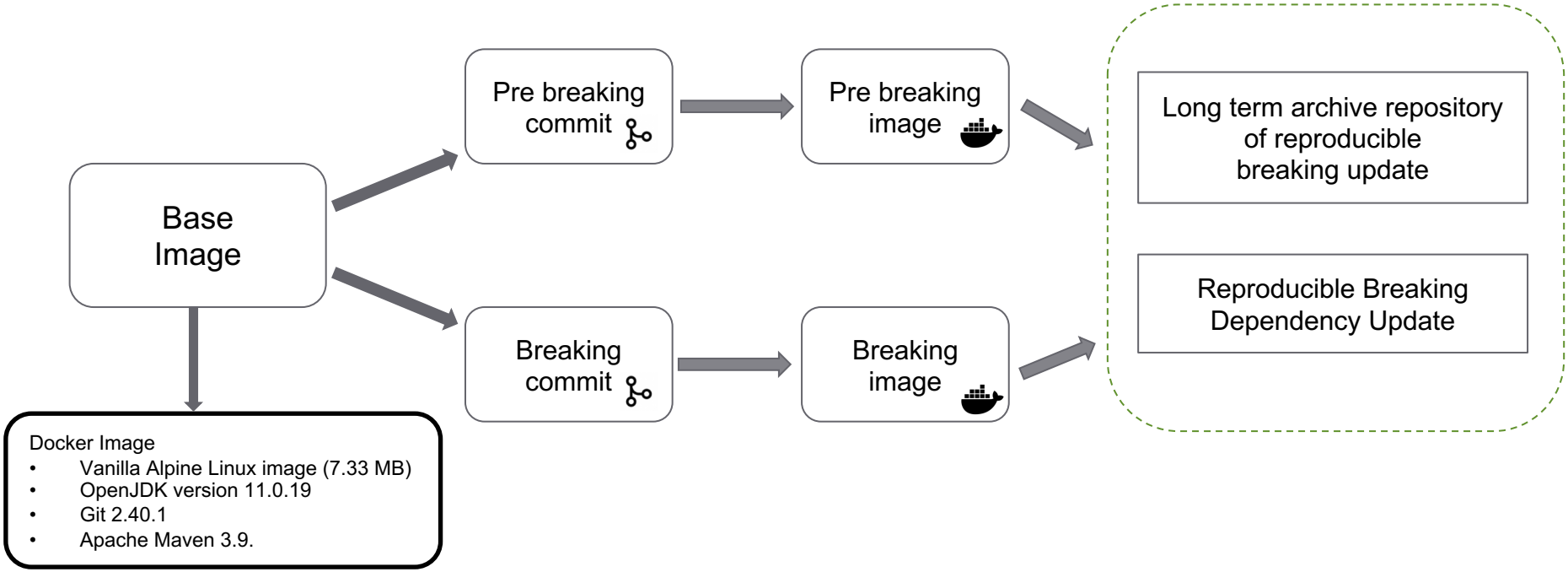
- Compilation failure: 243 (43%)
  - failure in the compilation phase
- Test failure: 188 (33%)
  - failures during test execution
- Enforcer failure: 121 (21%)
  - errors triggered by the dependency usage rule
- Dependency lock failure: 14 (2%)
  - project uses the dependency-lock-maven-plugin to lock versions of direct and transitive dependencies
- Dependency resolution failure: 5 (1%)
  - failures are related to missing artifacts

Compilation failure: 243(43%)

Most common causes for compilation failures

Causes	Total number of occurrences
cannot find symbol	**1950**
cannot access	**419**
package <> does not exist	**344**
constructor <> in class cannot be applied to given types	**129**
method <> cannot be applied to given types	**9**
no suitable constructor found for <>	**5**

Isolation of breaking updates



Conclusions

- BUMP, a benchmark of 571 breaking dependency updates collected from 153 Java projects on GitHub.

- BUMP stores the breaking updates as Docker image ensuring full reproducibility. https://zenodo.org/records/10041883.

- BUMP is the first benchmark of real-world, fully reproducible breaking dependency updates

BUMP: A Benchmark of Reproducible Breaking Dependency Updates

Frank Reyes, Yogya Gamage, Gabriel Skoglund, Benoit Baudry, Martin Monperrus

Thank you!

Questions?

BUMP repository

https://github.com/chains-project/bump

Research group

https://github.com/chains-project