

A Multi-Metric Ranking Approach for Library Migration Recommendations

Hao He¹, Yulin Xu¹, Yixiao Ma¹, Yifei Xu¹, Guangtai Liang², Minghui Zhou¹

¹Department of Computer Science, Peking University

²Software Analysis Lab, Huawei Technologies Co., Ltd.

Presenter: Hao He (何昊), Ph.D. Candidate



北京大学
PEKING UNIVERSITY



Background: The number of available 3rd-party libraries is rapidly growing



Library Count (by Package Manager)

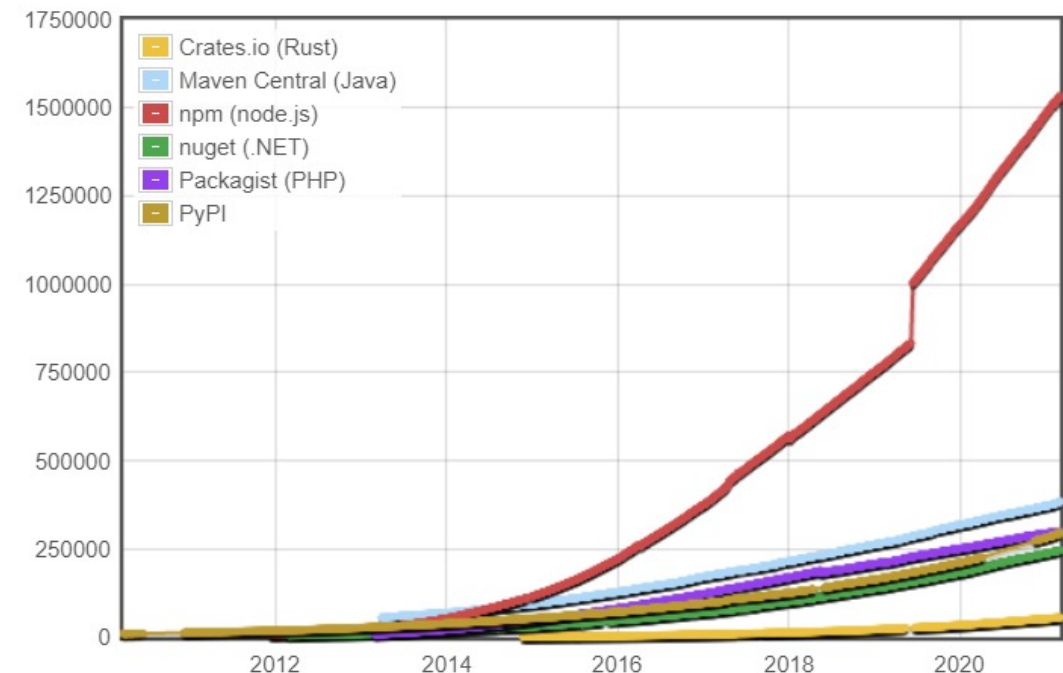


Image from <http://www.modulecounts.com/>

Maven Artifacts: ~50,000 in 2014, ~350,000 in 2021

Background: 3rd-party libraries are prone to failures, which trigger library migrations

- Security vulnerabilities
- Lack of maintenance
- License incompatibilities
- Misalignment with requirements

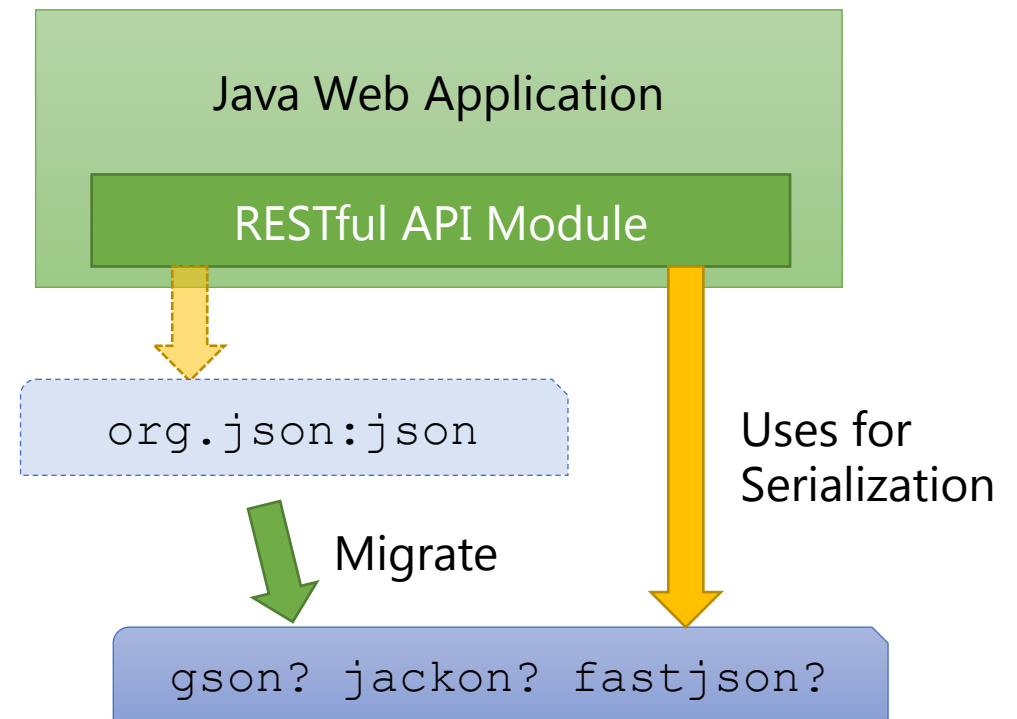


End of Life

On August 5, 2015 the Logging Services Project Management Committee announced that Log4j 1.x had reached end of life. For complete text of the announcement please see the [Apache Blog](#). Users of Log4j 1 are recommended to upgrade to [Apache Log4j 2](#).

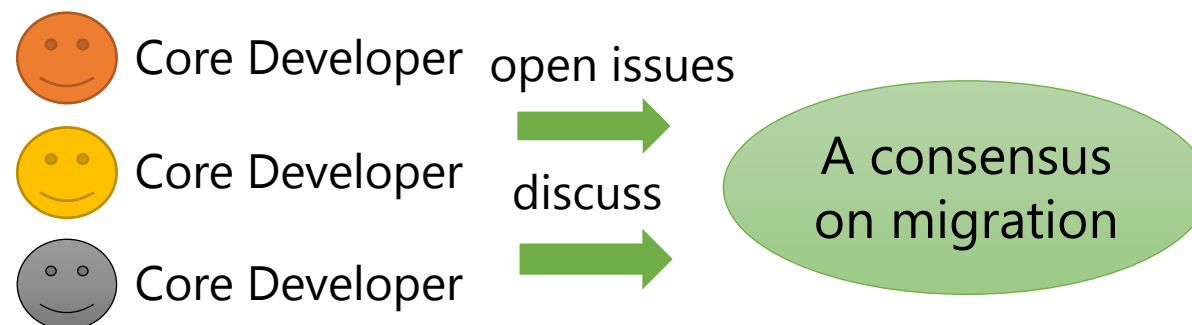


Scalability?
Performance?
Feature?
Security?



Motivation: Choosing the best library to migrate can be difficult...

- Most online resources are opinion-based discussions
- Developers make decisions based on prior experience



Typical open-source projects (Kabinna et al. [1])

1. Kabinna, Suhas, et al. "Logging library migrations: A case study for the Apache Software Foundation projects." *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. IEEE, 2016.

What if we can know what other
(high-quality) projects have done?

Problem

- **Input:** A library query a specified by user
- **Output:** A list of libraries to which the user can migrate from a
- **Constraint:** Use evidence from existing software repositories



User types a library name...

org.json:json Search



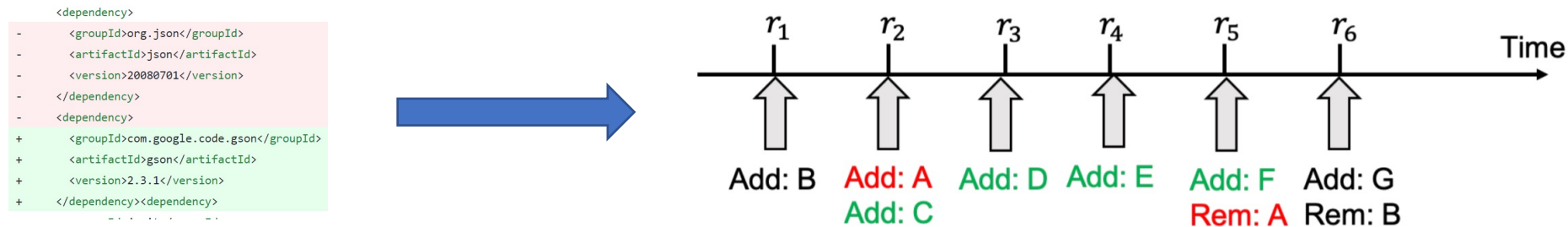
Returns

Rank	Target Library	Confidence
1	com.fasterxml.jackson.core:jackson-databind	0.2572
2	com.google.code.gson:gson	0.1669
3	com.alibaba:fastjson	0.0654

Repository	Commit
apache/asterixdb	54ecc63
release-engineering /pom-manipulation-ext	0f2ed00
opendaylight/aaa	315b173
.....

Existing Approaches in this Direction

- Mine library changes from revision history (commits or releases)



- Collect eligible target libraries for migration by **filtering**
 - **Teyton et al. [1]**: filter using change frequency in same revision
 - **Teyton et al. [2]**: filter using change frequency across all revisions
 - **Alrubaye et al. [3]**: filter using both library changes and API changes in source code

1. Teyton, Cedric, Jean-Remy Falleri, and Xavier Blanc. "Mining library migration graphs." 2012 19th Working Conference on Reverse Engineering. IEEE, 2012.
2. Teyton, Cédric, et al. "A study of library migrations in Java." Journal of Software: Evolution and Process 26.11 (2014): 1030-1052.
3. Alrubaye, Hussein, Mohamed Wiem Mkaouer, and Ali Ouni. "MigrationMiner: An automated detection tool of third-party Java library migration at the method level." 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2019. **(Note: we only compare its approach at library level)**

Our Insights

- Combine metrics to capture evidence from **multiple** sources

```
<dependency>
- <groupId>org.json</groupId>
- <artifactId>json</artifactId>
- <version>20080701</version>
- </dependency>
+ <dependency>
+ <groupId>com.google.code.gson</groupId>
+ <artifactId>gson</artifactId>
+ <version>2.3.1</version>
+ </dependency></dependency>
```

Library Changes

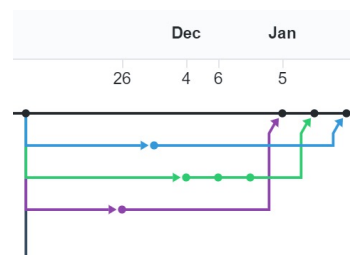
```
✓ replace JSON library from org.json
to com.google.code.gson.
master selenese-runner-java-3.33.0 ...
selenese-runner-java-2.4.0

vmi committed on 31 Jan 2016
1 parent 7837a97 commit 641ab94e7d014cdf4fd6i
```

Commit Messages

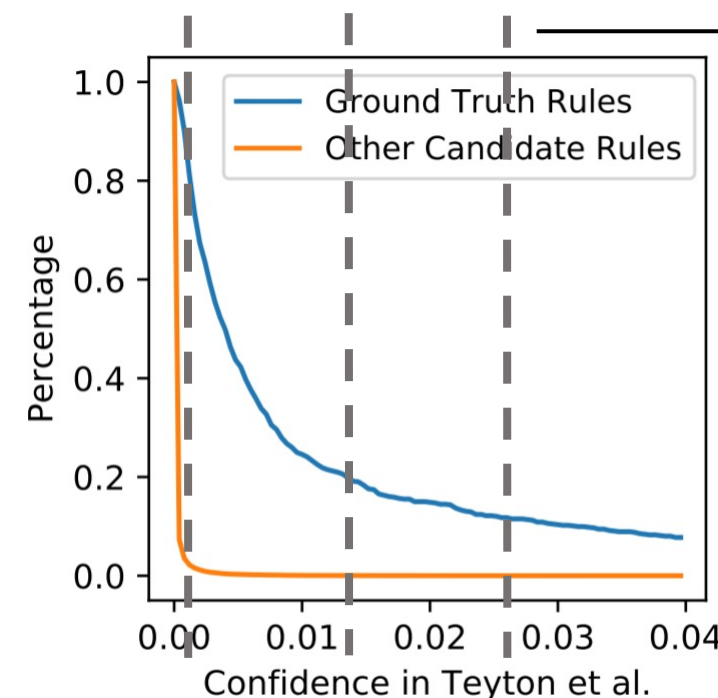
```
bindings.put("rule", ((JSMap<?, ?>) rule).unwrap());
else
bindings.put("rule", rule);
String args = new JSONObject(rollupArgs).toString();
String args = new Gson().toJson(rollupArgs);
try {
commands = JSList.toList(engine, engine.eval("rule.ge
} catch (ScriptException e) {
```

Source Code Changes



Commit Topology

- Rank** results of each library query by relative values



A global filtering threshold either:

- 1) Filter out many true positives
- 2) Keep many false positives

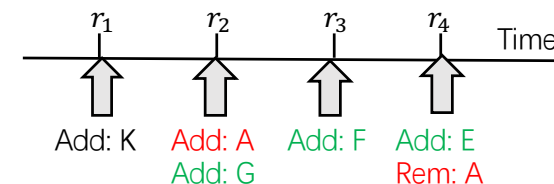
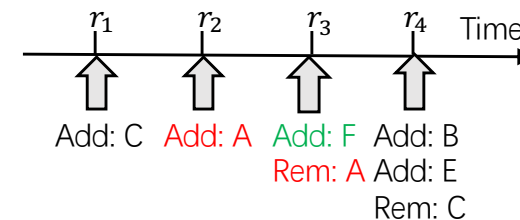
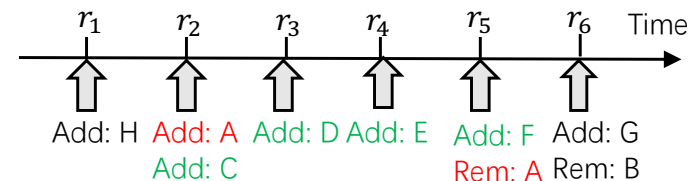
Rule Support (RS)

- **Intuition:** The frequently added libraries with the removal of a in the *same* commit are likely to be replacing a
- **Definition:**

$$RS(a, b) = \frac{RC(a, b)}{\max_{(a, x) \in R_c} RC(a, x)}$$

$RC(a, b)$ is the number of times library a is removed and library b is added in the same commit

- **Example (3 projects):**



$$RC(A, F) = 2$$

$$RC(A, E) = 1$$

$$RS(A, E) = 0.5$$

Message Support (MS)

- **Intuition:** Commit messages stating things like “migrate from a to b ” is direct evidence of an existing migration
- **Definition:**
- **Example**

$$MS(a, b) = \log_2(MC(a, b) + 1)$$

$MC(a, b)$ is the number of commits with commit message stating a migration from library a to library b



$$MC \left(\begin{matrix} org.json:json, \\ com.google.code.gson:gson \end{matrix} \right) += 1$$

API Support (AS)

- **Intuition:** Code changes that remove API references of a and add API references of b may be indicating a migration
- **Definition:**
- **Example**

$$AS(a, b) = \max(0.1, \frac{AC(a, b)}{\max_{(a, x) \in R_c} AC(a, x)})$$

$AC(a, b)$ is the number of hunks in which APIs of library a is added and APIs of library b is removed

```
@@ -49,7 +49,7 @@ public CommandList getExpandedCommands(Context context,
    bindings.put("rule", ((JSMap<?, ?>) rule).unwrap());
    else
        bindings.put("rule", rule);
-   String args = new JSONObject(rollupArgs).toString();
+   String args = new Gson().toJson(rollupArgs);
    try {
        commands = JSList.toList(engine, engine.eval("rule.getF
    } catch (ScriptException e) {
```

$$AC \left(\begin{matrix} org.json:json, \\ com.google.code.gson:gson \end{matrix} \right) += 1$$

Distance Support (DS)

• Intuitions:

- Previous metrics may miss migrations that span over multiple commits
- Migrations are unlikely to span over a large number of commits

• Definition:

$$DS(a, b) = \frac{1}{|Rev(a, b)|} \sum_{(r_i, r_j) \in Rev(a, b)} \frac{1}{(Dis(r_i, r_j) + 1)^2}$$

$Rev(a, b)$ is the set of revision pairs (r_i, r_j) that:

- 1) In the same project and r_i happens after r_j
- 2) Library a is removed in r_i
- 3) Library b is added in r_j

$Dis(r_i, r_j)$ is the number of commits that are committed between r_i and r_j

Confidence Value / Ranking Algorithm

- **Confidence Value Definition**

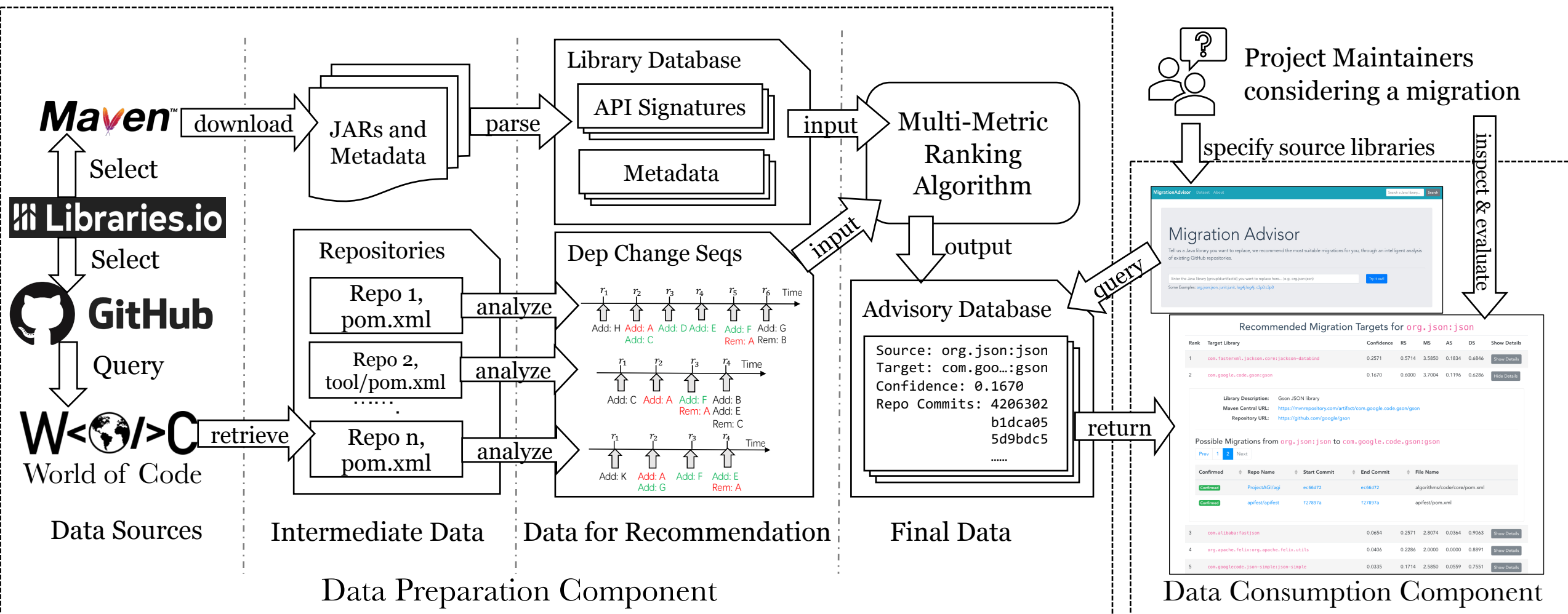
$$conf(a, b) = RS(a, b) \times MS(a, b) \times AS(a, b) \times DS(a, b)$$

- **Ranking Algorithm**

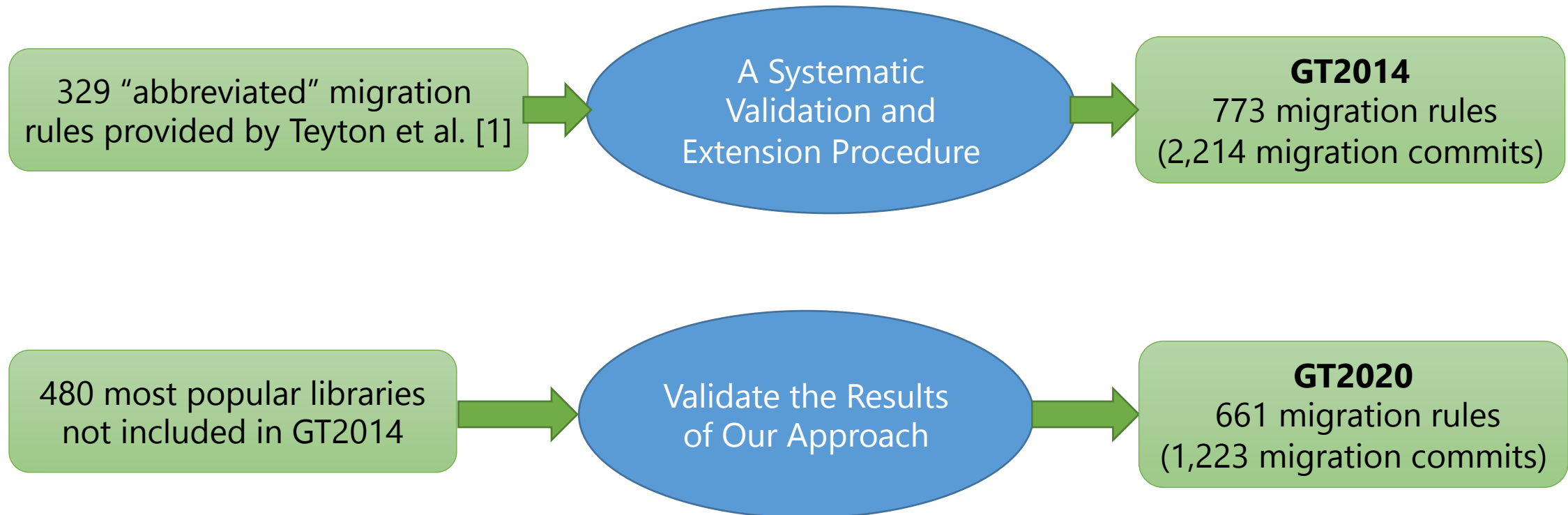
- Given library a , compute candidate library set C
- For $b \in C$, compute $conf(a, b)$
- Returns
 - C ranked by $conf(a, b)$
 - For $b \in C$, migration commits detected by $MC(a, b)$

Implementation

Demo Available: <http://migration-helper.net/>
 GitHub Repo: <https://github.com/hehao98/MigrationHelper>



Ground Truth – GT2014/GT2020



1. Teyton, Cédric, et al. "A study of library migrations in Java." Journal of Software: Evolution and Process 26.11 (2014): 1030-1052.

Performance Evaluation

Performance of Different Approaches using Ground Truth Dataset GT2014

		Approach	MRR	Precision@1	NDCG@10	Recall@20
Approaches from existing works	{	Teyton et al.	0.7133	0.6368	0.6056	0.7257
		Teyton et al.'	0.7335	0.6757	0.6909	0.6391
		Teyton et al."	0.8858	0.8737	0.8909	0.1759
		Alrubaye et al.	0.9412	0.9412	0.9412	0.0540
Other alternatives using only some of the proposed metrics	{	RS Only	0.7208	0.6474	0.6073	0.7270
		MS Only	0.7619	0.6737	0.6619	0.7736
		RS * MS	0.8275	0.7579	0.7436	0.8616
		RS * MS * DS	0.8401	0.7737	0.7479	0.8745
		RS * MS * AS	0.8379	0.7737	0.7479	0.8745
		Our Approach	0.8566	0.7947	0.7702	0.8939

Recommendation Example 1



User types a library name...

org.json:json

Search



Returns

Rank	Target Library	Confidence
1	com.fasterxml.jackson.core:jackson-databind	0.2572
2	com.google.code.gson:gson	0.1669
3	com.alibaba:fastjson	0.0654
4	org.apache.felix:org.apache.felix.utils	0.0406
5	com.googlecode.json-simple:json-simple	0.0334
6	org.apache.johnzon:johnzon-core	0.0331
7	com.fasterxml.jackson.core:jackson-core	0.0311
8	org.springframework.boot:spring-boot-starter-json	0.0265

Demo Available: <http://migration-helper.net/>

Repository	Commit
apache/asterixdb	54ecc63
release-engineering /pom-manipulation-ext	0f2ed00
opendaylight/aaa	315b173
.....

✓ Replace org.json with jackson Browse files

master (#198) pom-manipulation-parent-4.2 pom-manipulation-parent-1.7

VaclavDedik committed on 28 Aug 2015

1 parent 9f5b0e2 commit d7d4a0305e6b681e97e79b0510ca0b7570f2ed00

Showing 6 changed files with 149 additions and 71 deletions.

Unified Split

- io/pom.xml +7 -2
- io/src/main/java/org/commonjava/maven/ext/manip/rest/DefaultVersionTranslator.java +8 -43
- io/src/main/java/org/commonjava/maven/ext/manip/rest/mapper/ProjectVersionRefMapper.java +97 -0
- io/src/test/java/org/commonjava/maven/ext/manip/rest/DefaultVersionTranslatorTest.java +12 -10
- io/src/test/java/org/commonjava/maven/ext/manip/rest/handler/AddSuffixJettyHandler.java +17 -13
- pom.xml +8 -3

Recommendation Example 2



User types a library name...

c3p0:c3p0

Search



Returns

Demo Available: <http://migration-helper.net/>

Rank	Target Library	Confidence
1	com.mchange:c3p0	0.4083
2	com.zaxxer:HikariCP	0.0124
3	org.jboss.jbossts.jta:narayana-jta	0.0071
4	org.springframework.boot:spring-boot-starter-test	0.0050
5	com.alibaba:druid	0.0039
6	org.jboss.spec.javax.servlet:jboss-servlet-api_3.0_spec	0.0038
7	org.hibernate:hibernate-c3p0	0.0030
8	org.hibernate:hibernate-core	0.0029

Repository

Commit

kislayverma/Rulette

0832aaa

Moving data source to use HikariCP

Browse files

16545 committed on 1 Jan 2018

1 parent 9a447d3 commit 8664442927d5bf9fe9f0b8314453525d70832aaa

Showing 7 changed files with 38 additions and 38 deletions.

Unified Split

pom.xml

+1 -1

roulette-core/pom.xml

+1 -1

roulette-engine/pom.xml

+2 -2

roulette-examples/pom.xml

+3 -3

roulette-mysql-provider/pom.xml

+7 -2

roulette-mysql-provider/src/main/java/com/github/kislayverma/roulette/mysql/MySQLDataProvider.java

+3 -3

...ider/src/main/java/com/github/kislayverma/roulette/mysql/dao/{DataSource.java → MyDataSource.java}

+21 -26

Limitations

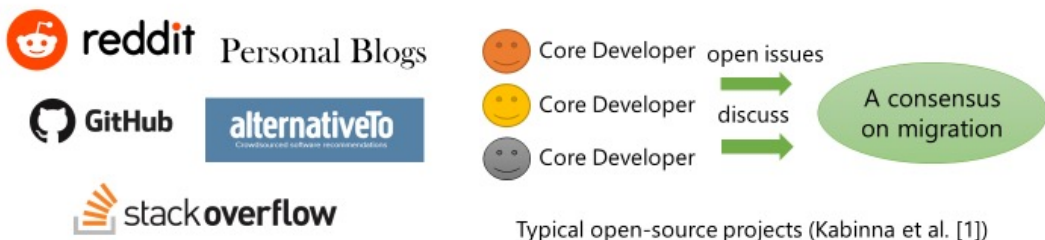
- Cold Start
- Data Sparsity
- Better Approaches

Open Problems

- How do the first migration happen?
- Reasons for migration?
- Why do projects migrate to different libraries?

Motivation: Choosing the best library to migrate can be difficult...

- Most online resources are opinion-based discussions
- Developers make decisions based on prior experience

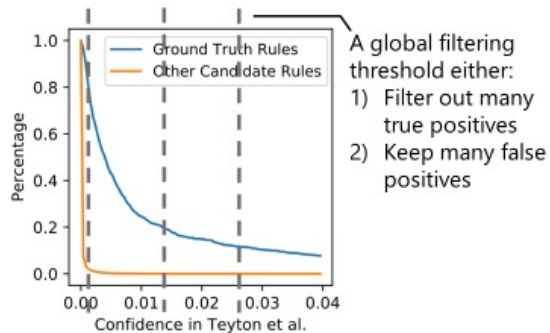
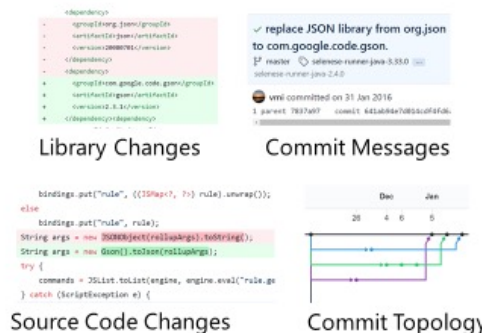


1. Kabinna, Suhas, et al. "Logging library migrations: A case study for the Apache Software Foundation projects." 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR). IEEE, 2016.

Contact: heh@pku.edu.cn, zhmh@pku.edu.cn
Thank You for your Attention!

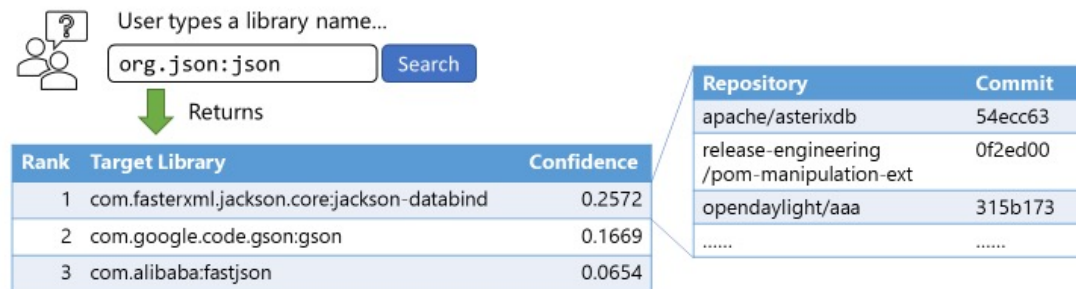
Our Insights

- Use metrics to capture evidence from **multiple** sources
- **Rank** results of each library query by relative values



Problem

- **Input:** A library query a specified by user
- **Output:** A list of libraries to which the user can migrate from a
- **Constraint:** Use evidence from existing software repositories



Performance Evaluation

Performance of Different Approaches using Ground Truth Dataset GT2014

		Approach	MRR	Precision@1	NDCG@10	Recall@20
Approaches from existing works	{	Teyton et al.	0.7133	0.6368	0.6056	0.7257
		Teyton et al.'	0.7335	0.6757	0.6909	0.6391
		Teyton et al."	0.8858	0.8737	0.8909	0.1759
		Alrubaye et al.	0.9412	0.9412	0.9412	0.0540
Other alternatives using only some of the proposed metrics	{	RS Only	0.7208	0.6474	0.6073	0.7270
		MS Only	0.7619	0.6737	0.6619	0.7736
		RS * MS	0.8275	0.7579	0.7436	0.8616
		RS * MS * DS	0.8401	0.7737	0.7479	0.8745
		RS * MS * AS	0.8379	0.7737	0.7479	0.8745
		Our Approach	0.8566	0.7947	0.7702	0.8939