JBoss
Community

Java Microbenchmarking Harness

OpenJDK

JBoss Community

About me

- Hardy Ferentschik - hardy@hibernate.org
- Hibernate team member
- Project lead of Hibernate Validator
- Bean Validation expert group member
- Interested in everything new and cool, e.g. Openshift, Docker, Kubernetes
- KIT Alumnus

Motivation

“The URL Validator uses the *java.net.URL* constructor to decide if an URL is valid. This is costly in terms of performance, since the URL class checks for the available URLStreamHandlers. This (a regular expression) is a much better and faster strategy: …”

— *HV-920*

```
public class User {
   @URL
   private String homepage;

   //…
}
```

It’s a regexp monster!

```
private static final Pattern URL_REGEX = Pattern
          .compile("(?i)^[a-z](?:[-a-z0-9\\+\\.])*:(?:\\/\\/(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\
\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\
\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:])*@)?(?:\\[(?:(?:(?:
[0-9a-f]{1,4}:){6}(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3})|::(?:[0-9a-f]{1,4}:)
{5}(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3})|(?:[0-9a-f]{1,4})?::(?:[0-9a-f]
{1,4}:){4}(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3})|(?:[0-9a-f]{1,4}:[0-9a-f]
{1,4})?::(?:[0-9a-f]{1,4}:){3}(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3})|(?:(?:
[0-9a-f]{1,4}:){0,2}[0-9a-f]{1,4})?::(?:[0-9a-f]{1,4}:){2}(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|
2[0-4][0-9]|25[0-5])){3})|(?:(?:[0-9a-f]{1,4}:){0,3}[0-9a-f]{1,4})?::[0-9a-f]{1,4}:(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|
[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3})|(?:(?:[0-9a-f]{1,4}:){0,4}[0-9a-f]{1,4})?::(?:[0-9a-f]{1,4}:[0-9a-f]{1,4}|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:
[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3})|(?:(?:[0-9a-f]{1,4}:){0,5}[0-9a-f]{1,4})?::[0-9a-f]{1,4}|(?:(?:[0-9a-f]{1,4}:){0,6}[0-9a-f]{1,4})?::)|v[0-9a-f]+[-a-z0-9\\._~!\\
$&'\\(\\)\\*\\+,;=:]+)\\]|(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(?:\\.(?:[0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3}|(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\
\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\
\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\
\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=@])*)(?::[0-9]*)?(?:\\/(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\
\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\
\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\
\)\\*\\+,;=:@]))*)*|\\/(?:(?:(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\
\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\
\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@]))+)(?:\\/(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\
\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\
\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\
\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@]))*)*)?|(?:(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\
\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\
\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\
\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@]))+)(?:\\/(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\
\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\
\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@]))*)*|(?!(?:%[0-9a-f][0-9a-f]|[-a-
z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-
\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\
\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@])))(?:\\?(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}\\x{F900}-\\x{FDCF}\
\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\\x{6FFFD}\\x{70000}-\
\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\\x{DFFFD}\\x{E1000}-\
\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@])|[\\x{E000}-\\x{F8FF}\\x{F0000}-\\x{FFFFD}|\\x{100000}-\\x{10FFFD}\\/\\?])*)?(?:\\#(?:(?:%[0-9a-f][0-9a-f]|[-a-z0-9\\._~\\x{A0}-\\x{D7FF}
\\x{F900}-\\x{FDCF}\\x{FDF0}-\\x{FFEF}\\x{10000}-\\x{1FFFD}\\x{20000}-\\x{2FFFD}\\x{30000}-\\x{3FFFD}\\x{40000}-\\x{4FFFD}\\x{50000}-\\x{5FFFD}\\x{60000}-\
\x{6FFFD}\\x{70000}-\\x{7FFFD}\\x{80000}-\\x{8FFFD}\\x{90000}-\\x{9FFFD}\\x{A0000}-\\x{AFFFD}\\x{B0000}-\\x{BFFFD}\\x{C0000}-\\x{CFFFD}\\x{D0000}-\
\x{DFFFD}\\x{E1000}-\\x{EFFFD}!\\$&'\\(\\)\\*\\+,;=:@])|[\\/\\?])*)?$");
```

How hard can it be!?

```
private static final int MAX_ELEMENTS = 10_000_000;

public static void main(String[] args) {
    List<? extends Set<Integer>> testees =
            Arrays.asList(
                    new HashSet<Integer>(),
                    new TreeSet<Integer>(),
                    new ConcurrentSkipListSet<Integer>());
    for (Set<Integer> testee : testees) {
        doBenchmark(testee);
    }
}
```

How hard can it be!?

```
private static void doBenchmark(Set<Integer> testee) {
    long start = System.currentTimeMillis();
    for (int i = 0; i < MAX_ELEMENTS; i++) {
        testee.add(i);
    }
    long end = System.currentTimeMillis();
    System.out.printf("%s took %d ms%n”,
        testee.getClass(), (end - start));
}
```

Problems

- Garbage collection
- JIT and On-Stack-Replacement (OSR)
- Dead code elimination
- System load

Enter JMH

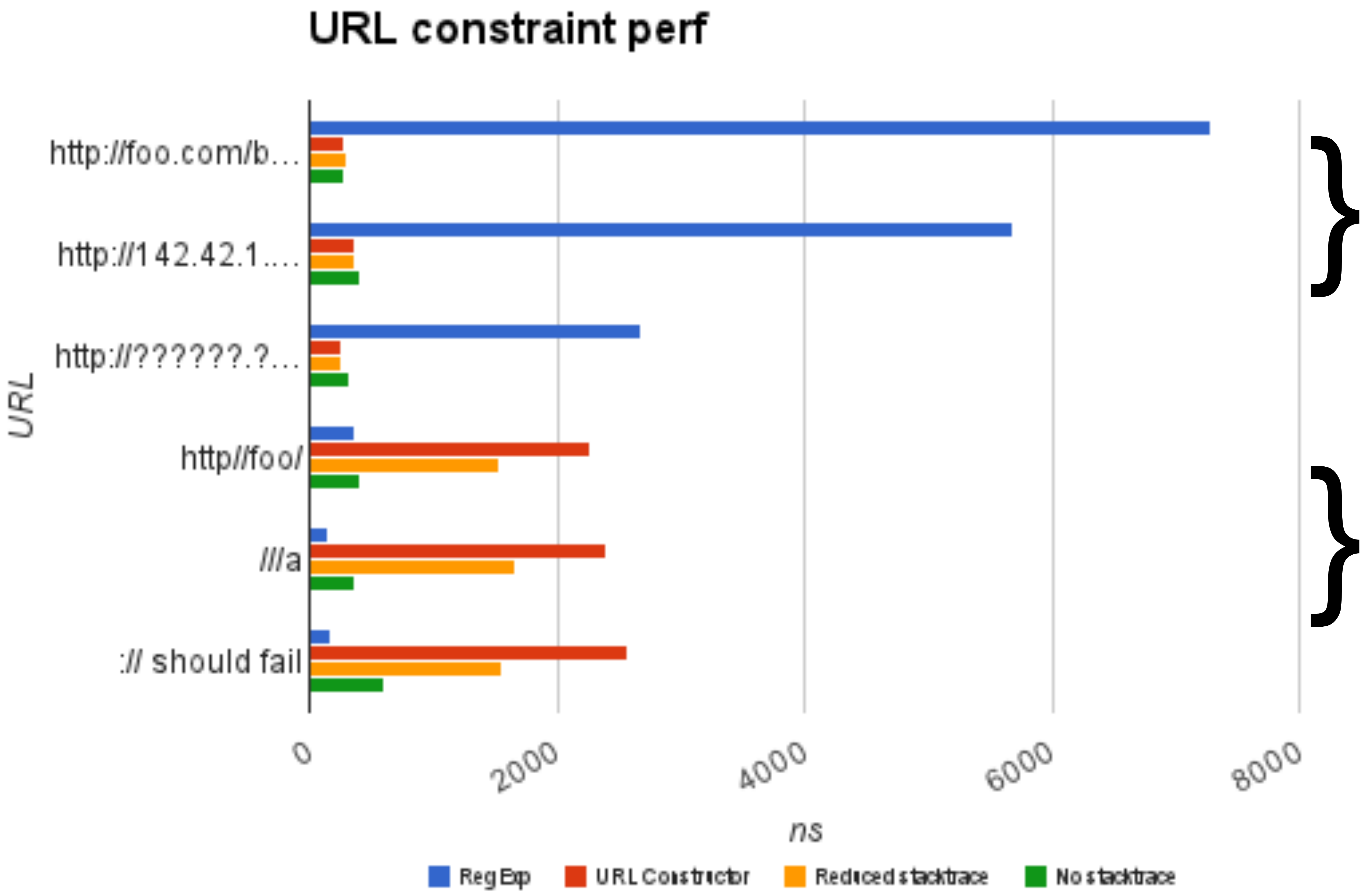
- Developed by Oracle/Sun Performance Engineering team
- Lead by Aleksey Shipilëv
- Roots in JRockit
- JMH 0.1 was released to the unsuspecting public in March 2013
- Current version 1.9.2

Getting started

```
mvn archetype:generate \
          -DinteractiveMode=false \
          -DarchetypeGroupId=org.openjdk.jmh \
          -DarchetypeArtifactId=jmh-simple-benchmark-archetype \
          -DgroupId=org.sample \
          -DartifactId=test \
          -Dversion=1.0
```

maven

Demo



} valid URLs

} invalid URLs

Summary

- Annotation driven
- Annotation Processors (JSR 269) create benchmark classes
- Run via shaded jar file or as executing the Runner directly
- Configure via annotations, fluent API or command lines flags
- JMH Samples best place to start

Links

- http://openjdk.java.net/projects/code-tools/jmh
- hg.openjdk.java.net/code-tools/jmh/file/tip/jmh-samples/src/main/java/org/openjdk/jmh/samples/
- http://daniel.mitterdorfer.name/categories/series-jmh-intro
- http://psy-lob-saw.blogspot.se/p/jmh-related-posts.html

https://github.com/hferentschik/jmh-perf-tests

user:hferentschik

Q + A