Университет ИТМО, факультет программной инженерии и компьютерной техники
Двухнедельная отчётная работа по «Информатике»: аннотация к статье
Дата прошедшей лекции: 11.10.2022 Номер прошедшей лекции: №3 Дата сдачи: 25.10.2022

Выполнил(а) Барсуков М. А., № группы *P3115*, оценка _____
Фамилия И.О. студента — не заполнять

Название статьи/главы книги/видеолекции
Еще раз о регекспах, бэктрекинге и том, как можно положить на лопатки JVM двумя строками «безобидного» кода

ФИО автора статьи (или e-mail)	**Дата публикации (не старше 2019 года)**	**Размер статьи (от 400 слов)**
@Starcounter	"28" февраля 2021 г.	1239

Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)
https://habr.com/ru/company/vk/blog/544688/

Теги, ключевые слова или словосочетания
Регулярные выражения, backtracking, ReDoS, RE2, JVM

Перечень фактов, упомянутых в статье
1. Иногда вычисление регекспа на небольшой строке может выполняться очень долго.
2. Бэктрекинг в регулярных выражениях – жадный алгоритм, катастрофический бэктрекинг означает полный перебор, т.е. время выполнения будет расти экспоненциально.
3. Причиной катастрофического бэктрекинга зачастую является неэффективный регексп.
4. ReDos-атака (Regular expression Denial of Service attack) приводит к «отказу в обслуживании» из-за ввода данных, для оценки которых регексп тратит много времени.
5. В JDK >9 производительность стандартной библиотеки для регекспов значительно улучшилась.
6. RE2 и RE2/J – библиотеки для регулярных выражений, которые по производительности выше Perl-совместимых, однако у них есть много минусов: синтаксис регекспов часто не совпадает, основным мейнтейнером является единственных разработчик и т.д.

Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)
1. Во многих языках существуют библиотеки для регекспов, позволяющие производить матчинг за линейное время.
2. Средства для работы с регулярными выражениями все время совершенствуются и становиться более производительными.
3. Существует множество программ, позволяющих проверить регекспы с катастрофическим бэктрекингом (например, vuln-regex-detector для Perl или safe-regex для Node.js).

Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)
1. Невнимательно написанные регулярные выражения часто являются уязвимыми для ReDos-атаки.
2. В движках регулярных выражений почти невозможно сочетать стабильность, функциональность и производительность.
3. Сторонние решения, поддерживающие матчинг за линейное время, часто требуют переписывания регулярного выражения с нуля.

Ваши замечания, пожелания преподавателю *или* анекдот о программистах

Even Jon Skeet cannot parse HTML using regular expressions. Every time you attempt to parse HTML with regular expressions, the unholy child weeps the blood of virgins, and Russian hackers pwn your webapp. Parsing HTML with regex summons tainted souls into the realm of the living. HTML and regex go together like love, marriage, and ritual infanticide. The cannot hold it is too late. The force of regex and HTML together in the same conceptual space will destroy your mind like so much watery putty. If you parse HTML with regex you are giving in to Them and their blasphemous ways which doom us all to inhuman toil for the One whose Name cannot be expressed in the Basic Multilingual Plane, he comes.

(© https://stackoverflow.com/questions/1732348, 2009)