

Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра Системного Программирования

Кладов Алексей Александрович

Разработка системы проверки упражнений для образовательной платформы

Дипломная работа

Допущена к защите.
Зав. кафедрой:

Научный руководитель:

Рецензент:
Вяххи Н. И.

Санкт-Петербург
2014

SAINT-PETERSBURG STATE UNIVERSITY
Mathematics & Mechanics Faculty

Chair of Software Engineering

Aleksei Kladov

Quiz checking system for educational engine

Graduation Thesis

Admitted for defence.
Head of the chair:
professor Terehov Nikolay

Scientific supervisor:

Reviewer:
Vyahhi Nikolay

Saint-Petersburg
2014

Оглавление

Введение	4
1. Платформа Stepic	6
2. Постановка задачи	8
3. API для создания новых типов упражнений	9
3.1. Детали Реализации	10
4. Реализованные типы упражнений	11
5. Изолированное исполнение кода упражнений	13
5.1. Изоляция	13
5.2. Масштабирования	13
Заключение	14

Введение

В последнее время технологии Интернет активно используются чтобы улучшить качество предоставляемого образования. Комплекс технологий и программ, осуществляющих такое улучшение, получил название электронного обучения [<http://www.eurodl.org/ma>]

Особенности электронного обучения

Электронное обучение отличается некоторыми особенностями.

Во-первых, это большое количество студентов в одном потоке, как правило достигающее нескольких десятков тысяч человек.

Во-вторых, хотя учебные материалы уникальны для каждого курса, они, как правило, очень высокого качества – видео лекции, субтитры, аннотированные слайды, книги.

В-третьих, мотивация студентов курса зачастую не очень высокая. Это связано с тем, что записаться на Интернет курс значительно проще, чем поступить в университет.

В-четвёртых, свободный формат электронного обучения позволяет учиться в удобное для студента время.

История

Пожалуй, история компьютерного образования началась с началом распространения персональных компьютеров, если не раньше, однако можно выделить следующие вехи:

Walter Lewin – профессор физики из MIT, лекции которого показывались по местному телевидению

open courseware – MIT сделал все учебные материалы доступными в Интернет.

Udacity – первый стартап, с MOOC в настоящем его понимании.

Coursera – наиболее успешная на настоящий момент платформа для online образования

Проблемы

online образованию присущи и некоторые специфичные проблемы. Среди них можно выделить следующие.

- Ограничены возможности проверки знаний. Так как упражнения должны проверяться автоматически, то их набор часто ограничен.

- Большая стоимость создания online курса. Создание курса требует значительных затрат на запись видео лекций, оформление электронного конспекта и создание набора упражнений.
- Drop out. Для online образования характерен значительно меньший процент заканчивающих курс.

Стоит заметить, что в электронном образовании лекционный материал может быть представлен очень хорошо, зачастую лучше, чем в похожем классическом обучении. Так как студент сам активно взаимодействует с лекционным материалом, он может изучать его с удобной для себя скоростью. Можно сказать, что лекции хорошо “масштабируются” по количеству студентов. Но при этом практическая часть обучения – решение задач и выполнение упражнений – в электронном обучении реализуется значительно хуже, так как нет возможности проверки заданий студентов учителем.

Существующие платформы

*** Udacity 2011 год 1.6 млн *** Coursera 2012 год 7.1 млн *** edX 2012 год 2.1 млн MOOC.org

1. Платформа Stepic

Статус проекта

Stepic это молодой проект, развивающийся в рамках компании JetBrains. Разработка проекта стартовала в 2013 году. На текущий момент в проекте занято 8 человек.

Использование

На текущий момент Stepic используют 23 тысячи студентов. На платформе создано 400 уроков, многие из которых объединены в курсы.

Из завершившихся курсов стоит отметить следующие.

”Алгоритмы в биоинформатике”

Этот англоязычный курс проходил одновременно с соответствующим курсом на coursera, он состоял из большого количества текстовых материалов и упражнений на программирование из области алгоритмической биологии. В этом курсе приняли участие более 10 тысяч человек со всего мира.

”Алгоритмы и Структуры Данных”

Это закрытый на настоящий момент курс, который использовался для предварительной оценки знаний абитуриентов Computer Science Center в Санкт-Петербурге. В нём приняли участие 500 человек из Петербурга. Курс состоит из видео лекций на русском языке. Для курса использовались разные типы упражнений, но наиболее часто упражнения на программирование.

Возможности платформы

Stepic ориентирован на интеграцию и сотрудничество с другими инструментами online образования. Для этого поддерживаются стандарты oEmbed и LTI.

Единицей учебного материала на Stepic является урок – набор упражнений и/или теории представленный в виде слайдов, общим количеством не превосходящий 16 штук. Уроки можно объединять в курсы.

Курс позволяет создавать секции уроков, назначать разным упражнениям разные стоимости, создавать коллективы студентов и преподавателей, устанавливать предельные сроки сдачи упражнений.

Теория может быть представлена в виде html слайдов, или в виде видео лекций.

Необходимость расширять возможности автоматически проверяемых упражнений.

Технологии и Инструменты

В качестве базы данных используется MySQL. Разработка серверной части ведётся на языке Python 3, с использованием фреймворка django. Клиентская часть

разрабатывается на CoffeeScript, с использованием Ember.js.

Также используются celery для распределённого выполнения заданий, codejail в качестве основы системы изолированного исполнения кода, flask тоже зачем-то используется.

2. Постановка задачи

Целью работы является реализация системы для создания и проверки упражнений для образовательной платформы Stepic, с возможностью легко добавлять новые типы упражнений, в том числе и сторонним разработчикам.

Для достижения этой цели были сформулированы следующие задачи.

- Обеспечить возможность лёгкого расширения набора типов упражнений сторонними разработчиками (реализовать соответствующий API к платформе Stepic) и протестировать на практике его удобство.
- Реализовать с помощью разработанного API в Stepic типы упражнений, часто встречающихся в других образовательных платформах и проверить их работу на практике.
- Реализовать возможность масштабирования и изолированного исполнения потенциально не безопасного кода упражнений.

3. API для создания новых типов упражнений

Взаимодействие пользователя с упражнением в общем виде можно описать следующим образом. Сначала пользователь читает условие упражнения. Затем он нажимает на кнопку "начать решать". После этого пользователю представлен один из вариантов входных данных. Пользователь взаимодействует с клиентской частью упражнения и составляет свой ответ. Когда ответ готов, пользователь нажимает на кнопку отправить, после чего ответ проверяется. После проверки пользователь видит результат (бинарное верно/неверно или оценка – вещественное число от 0 до 1) и возможный текстовый отзыв.

Входные данные отличаются от попытки к попытке и генерируется случайным образом на сервере. Вместе с входными данными создаётся ключ к решению – объект, с помощью которого можно быстро проверить ответ студента.

Генерация пар входные данные/ключ и проверка ответа пользователя происходят асинхронно. При этом пара вход/ключ создаётся заранее, и ключ выбирается таким образом, чтобы проверка ответа выполнялась быстро. Таким образом, большую часть работы можно выполнить заранее, и ускорить получение студентом результата.

Вызовы API

В результате анализа различных типов упражнений был выявлен следующий общий набор операций, необходимых для создания любого упражнения.

- Создание пары входные данные/ключ.
- Отображение входных данных для пользователя.
- Получение ответа пользователя
- Проверка ответа пользователя при помощи ключа, оценка вещественным числом из интервала $[0; 1]$ и генерация текстовой подсказки.

Создание входных данных и проверка ответа могут быть долгими операциями. Поэтому, чтобы минимизировать время ожидания пользователя, пары входные данные/ключ генерируются заранее, и ключ подбирается таким образом, чтобы проверка ответа занимала как можно меньше времени.

Система Модулей

Каждый тип упражнения представляет собой отдельный модуль, никак не связанный с остальной платформой. Коллекция модулей доступна в публичном репозитории на github. Чтобы добавить новый модуль, автору необходимо сделать pull request в репозиторий.

Архитектура Решения

Упражнения работают по модели клиент-сервер.

Серверная часть обеспечивает создание входных данных и проверку ответов. Теоретически, её можно было бы осуществлять и на клиенте, но в таком случае было бы просто подделать сдачу упражнения.

Клиентская часть работает в браузере и обеспечивает богатое взаимодействие с пользователем. Клиент общается с сервером при помощи ајах запросов и упражнения работают без перезагрузки страницы.

3.1. Детали Реализации

Все данные упражнения сохраняются в формате JSON. Коммуникация между клиентом и сервером также происходит в формате JSON.

Серверная часть модуля должна быть написана на Python. Клиентская часть может быть написана на javascript или coffescript, с необязательным использованием ember.

Сервер Для Разработки

Для упрощения разработки модулей упражнений был написан сервер, который позволяет запускать упражнения без Stepic. Сервер автоматически перезагружается при изменении исходного кода упражнений.

4. Реализованные типы упражнений

С помощью API в Stepic были разработаны следующие типы упражнений:

- Choice
- Code
- Dataset
- Free Answer
- Math
- Number
- Sorting
- String

Примеры Конкретных Упражнений

Рассмотрим некоторые из них более подробно.

- choice
- dataset
- code

Choice просит студента выбрать среди предложенных вариантов ответа правильные. У этого квиза есть много опций – размер выборки, её рандомизация, количество правильных ответов в выборке. Это наиболее часто встречающийся в различных платформах тип квиза из-за своей простоты. Однако у него есть свои недостатки – например, студент может просто угадывать ответы. И студент не может придумать своё, по настоящему оригинальное решение

Dataset предлагает скачать текстовые входные данные и требует текстового ответа. Этот тип упражнений подходит для некоторых упражнений на программирование. Он хорош тем, что позволяет использовать любой инструмент для решения. Но у этого типа есть и недостатки. Размер входных данных не может быть большим, чтобы их удобно было скачивать через Интернет. Сложно проконтролировать эффективность пользовательского решения, студенту необходимо иметь нужные компиляторы на своей машине.

Code дополняет dataset. В этом упражнении студент должен послать код для решения задачи, который затем будет исполнен на сервере, с замерами времени и памяти. Это удобно для проверки эффективности решения, однако необходимо решать

задачу на одном из поддерживаемых языков программирования(Java, Python, C++, Octave).

Использование Упражнений на Практике

Созданные упражнения были успешно использованы на практике, при этом в разных курсах использовались разные типы упражнений.

Так в курсе "Алгоритмы в биоинформатике" большую часть упражнений составляли dataset упражнения. Этот тип упражнений оказался наиболее удобен, так как позволяет делать задачи про обработку больших объёмов данных, что характерно для биоинформатики, используя при этом любой язык программирования.

А курсе "Алгоритмы и структуры данных" большую часть упражнений составляли code и free answer упражнения. Code quizzes оказались удобны, так как позволяют ограничить решения по времени и памяти, что необходимо для проверки эффективности реализации алгоритмов. Free answer упражнения использовались для проверки теоретических задач.

5. Изолированное исполнение кода упражнений

Для многих важных типов упражнений необходимо исполнять потенциально небезопасный код. Из упражнений, реализованных в Stepic, такими являются следующие.

`dataset` – в этом упражнении входные данные генерируются при помощи кода на питоне, который вводит пользователь.

`code` – в этом типе упражнений ответом является код программы на Python, C++, java, haskell или octave.

`math` – в этом упражнении необходимо вычислять произвольное математическое выражение, также записанное в синтаксисе Python

`string` – это упражнение позволяет задавать для проверки регулярное выражение. Во многих современных языках программирования проверка строки на соответствие регулярному выражению занимает экспоненциальное время, что может привести к DOS атаке.

5.1. Изоляция

Для изоляции кода был существенно расширен codejail. Добавлена поддержка различных языков программирования, улучшены сообщения об ошибках, созданы профили armor для java, octave, haskell и C++.

5.2. Масштабирования

celery, nuff said

Заключение

В результате проделанной работы все поставленные цели были достигнуты.

Были разработаны следующие типы упражнений: choice, code, dataset, free answer, math, number, sorting, string. Эти типы упражнений были успешно использованы в курсах “Алгоритмы в Биоинформатике”, “Алгоритмы и Структуры Данных” и других.

Был разработан API для создания новых типов упражнений в виде подключаемых модулей. API был документирован. Для облегчения разработки был создан небольшой сервер для тестирования модулей. С использованием API был создан первый сторонний модуль упражнения. Существующие модули были опубликованы на репозитории на GitHub.

На основе code jail была создана система безопасного исполнения кода с возможностью ограничивать и измерять затраченные ресурсы. При помощи celery достигнуто масштабируемое и распределённое исполнение кода упражнений.