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:
д. ф.-м. н., профессор Хунта К. Х.

Научный руководитель:
д. ф.-м. н., профессор Выбегалло А. А.

Рецензент:
ст. преп. Вяххи Н. И.

Санкт-Петербург
2014

SAINT-PETERSBURG STATE UNIVERSITY
Mathematics & Mechanics Faculty

Chair of The Meaning of Life

Edelweis Mashkin

Empty subset as closed set

Graduation Thesis

Admitted for defence.
Head of the chair:
professor Christobal Junta

Scientific supervisor:
professor Amvrosy Vibegallo

Reviewer:
assistant Alexander Privalov

Saint-Petersburg
2014

Оглавление

Введение

Научно технический прогресс стремительно меняет все сферы современной жизни. В последнее время технологии Интернет активно используются чтобы улучшить качество предоставляемого образования. Комплекс технологий и программ, осуществляющих такое улучшение, получил название online образования.

Особенности online образования

online образование 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, аннотированые слайды, книги.

В-третьих, мотивация студентов курса зачастую не очень высокая.

В-четвёртых, свободный формат online образования позволяет учится в удобное для студента время.

История

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

Walter Lewin – проффессор 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 проверяются 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.

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 н человек.

Использование

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

Из завершившихся курсов стоит отметить следующие.

"Алгоритмы в биоинформатике"

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

"Алгоритмы и Структуры Данных"

Это зарытый 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 состоял 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 сладов, 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 представленна в виде html слайдов, или в виде видео лекций.

Необходимость расширять возможности автоматически проверяемых упражнений.

Технологи и Инструмены

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.

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

3. Реализованные типы упражнений

Общий Вид Упражнения

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.

Примеры Конкретных Упражнений

Для примера рассмотрим следующие типы упражнений

- 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 подходить 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 проконтроллировать эффективность пользовательского решения, студенту необходимо иметь нужные компиляторы на своей машине.

Code дополняет dataset. В этом упражнении студент должен послать код для решения задачи, которы затем будет исполнен на сервере, с замерами времени и памяти. Это удобно для проврки эффективности решения, однако необходимо решать задачу на одном из поддерживаемых языков программирования(Java, Python, C++, Octave).

На текущий моемнт, в Stepic реализованы следующие типы упражнений:

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

Использование Упражнений в Реальных Курсах

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

В курсе "Алгоритмы и структуры данных" большую часть упражнений составляли code quizы и free answer quizы. Code quizы оказались удобны, так как позволяют ограничить решения по времени и памяти, что необходимо для курса по алгоритмам. Free answer квизы использовались для проверки теоретических задач.

4. API для создания новых типов упражнений

Система Модулей

Так как для некоторых курсов необходимы спецефические типы упражнений, одним из требований подсистемы упражнений является возможность расширения набора упражнений сторонними разработчиками – авторами курсов и их ассистентами.

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-серевер.

Серверная часть обеспечивает проверку упражнений. 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 серевером при помощи ajax запросов и упражнения работают без перезагрузги страницы.

Сервер упражнения предоставляет следующий API:

- создания или изменение экземпляра упражнения
- получение dataset для упражнения
- отправка решения

Данные упражнения сохраняются в виде JSON. Коммуникация между клиентом и сервером также происходит в формате JSON.

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

Возможности API

trololo

Сервер Для Разработки

Для упрощения разработки плагинов написан сервер, позволяющий проверить работу плагина без

5. Изолированное исполнение кода упражнений

Для некоторых типов упражнений требуется возможность исполнять код изолированного и с ограничением используемых ресурсов. Например это необходимо для dataset, и code.

Интересно, что эта функциональность используется и в string. Дело в том, что в Python, как и в большинстве других популярных языков программирования, проверка строки на соответствие регулярному выражению может занимать экспоненциально время.

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 использованны 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 бы 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 масшабируемое и распредлённое исполнение кода упражнений.