Developing a Data-Driven Learning Interest Recommendation System to Promoting Self-Paced Learning on MOOCs

Prepared by Hsuan-Ming Chang

Directed by Prof. Nen-Fu Huang

In Partial Fulfillment of the Requirements

for the Degree of
Master of Science

Department of Computer Science

National Tsing Hua University

Hsinchu, Taiwan 30013, R.O.C.

E-mail: robert-rino@gapp.nthu.edu.tw

June 10, 2016

Abstract

This work proposes a learning-based energy management policy that takes into consideration the trade-off between the depth-of-discharge (DoD) and the lifetime of batteries. The impact of DoD on the energy management policy is often neglected in the past due to the inability to model its effect on the marginal cost per battery usage. In this work, a novel battery cost evaluation method that takes into consideration the DoD of each battery usage is proposed, and is utilized to devise the day-ahead energy management policy using reinforcement learning and linear value-function approximations. The policy determines the amount of energy to purchase for the next day in the day ahead market. A least-square policy iteration (LSPI) with linear approximations of the value function is used to learn the energy management policy. Simulations are provided based on real load profiles, pricing data, and renewable energy arrival statistics. The consideration of the battery cost due to DoD provides a more accurate evaluation of the actual energy cost and leads to an improved energy management policy.

Keywords -Smart grid, energy management system, reinforcement learning, battery, energy storage, depth-of-discharge.

This work was supported in part by the MOST R.O.C. Project on "Demand-Side Management in Universities and Enterprise Campuses".

Contents

Chapter 1

Introduction

Massive Online Open Courses (MOOCs) refer to an open educational resources, which allows learners worldwide to take well-designed online courses of interest free of charge. On MOOCs, learners watch the high-quality instructional videos made by professors from prestigious universities, share their ideas and reflections on the discussion forum, and use the online exercise system to evaluate their learning outcome. Due to the fact that the MOOCs provide with high-quality self-directed learning environment without costing much for online learners, MOOCs have been thought of as a contemporary way of 21-century learning.

There are two type of MOOCs, cMOOCs (connectionism MOOCs) and xMOOCs (instructionism MOOCs). These two types of MOOCs are base on different philosophical positions underpinning, cMOOCs focuses on connections between participants in particular on strong content contributions from the participants themselves [1], xMOOCs, by contrast focus on instructor's design of the course. Many famous MOOCs platform such as Coursera[2], edX[3], and Udacity[4] are belong to xMOOCs.

For current xMOOCs, the instructional videos play a significant role in the online learning process [5,6]. In essence, the learning focuses in the form of visual and

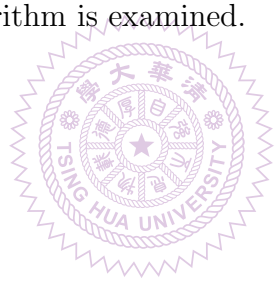
audial presented in the instructional videos. Traditionally, video-based learning follows structured instructor-designed sequences for the better results. Owing to the technological nature of the online stream video, it is found that many students drag the play bar replaying specific concept in the video for consolidating their understanding. Therefore, many studies aim to improve the video-based learning environment by adding additional features in video-watching, such as embedded assessment, caption tool, as so on.

In view of the rapid development of data sciences, more and more studies on educational data mining and learning analytics take the advantages of the learners data to optimize learning process. For example, [7] develops a step-by-step annotations feature to improve the learning experience of existing how-to videos. Study [8] constructs a system that recommends students videos best on their forum post, making a self-solved confusion system and meanwhile reducing the teaching load. Therefore, considering the learning needs and the authentic learner data, this study develops a data-driven learning interest recommendation system to promote self-paced learning by integrating educational data mining and word segmentation in the Chinese-speaking MOOC environment. Videomark combines both the learning seek event counts and the subtitles of each video to automatically generate learning concepts for learners in friendly user interface. Through the huge amount of video watching/seeking log data, the Videomark helps learners to quickly identify popular video seek events for consolidating their concept of the learning focus in hope of promoting better self-paced video-based learning environment.

This thesis proposes a learning interest recommendation system on xMOOCs, the system generate keywords relatevive to lecture video content base on students' video watch activity records and lecture video transcripts. Moreover, each keyword

collects video segment about the specific keyword. We hope this system will help students sketch the course when they first come to the class, and review the whole course after the course.

The remainder of this thesis is organized as follows. In Section **??**, we first introduce the Markov decision process and brief review the reinforcement learning algorithm which is called least-square policy iteration. In Section **??**, the energy management problem at the consumer side is examined. We designed our system model by considering a EMS center which want to regulate energy flow such as day-ahead energy purchasing, real-time energy purchasing, and energy dumping to minimize marginal cost and prolong battery life. In Section **??**, the reinforcement learning based energy management problem is examined. In Section **??**, the performance of purposed algorithm is examined.



Chapter 2

Related Work

2.1 MOOCs

2.1.1 Overview

MOOCs refers to massive online open courses, which is a online course platform people can access through internet connection regardless of the limit of space and time. MOOCs is normally free, credit-less and is designed for massive people to enroll and learn. Base on different theories, there are two kinds of MOOCs, cMOOCs and xMOOCs, which will mentioned in following section.

2.1.2 cMOOCs v.s. xMOOCs

Since "The year of MOOCs" [9], MOOCs have become increasingly robust and diverse in last few year. There are many websites provide MOOCs for different group in various way. Based on different learning theory, we classify MOOCs into cMOOCs and xMOOCs. cMOOCs are base on the learning theory of Connectivism, this kind of MOOCs focus on network between individuals in course. students may use any digitle platform such as Facebook, Google+, blog to make connections with other learners to create and construct knowledge. The partic-

ipants in cMOOCs act as teacher and learner at the same time as they share knoledge with each other. Instead of being structured as an open online community of learners, xMOOCs are much more like traditional classroom environment. xMOOCs are centered around class instroctor, instroctor usually will provide series of lecture video, where learners mainly get knowledges. Besides, exercises, quizes, assignments are also used during the course. Most of the popular MOOC websites in recent year are belong to xMOOCs such as Coursera, edX, Udacity, etc.

2.1.3 Coursera

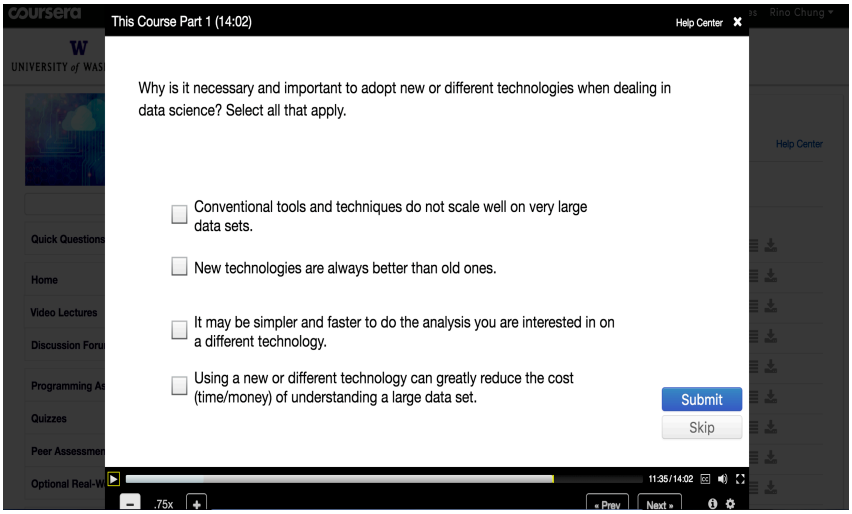


Figure 2.1: Coursera pop-up question in lecture video

Coursera [2] started in 2012 founded by Stanford University. It provides courses from renowned universitise like Stanford University, Princeton University, University of Michigan, etc. Taking course on Course is free, learners can enroll courses that interest them at will. Applying for hard copy certification of course, however,

will charge for some money. As a xMOOC platform, lecture videos are the major teaching material of courses with transcripts in many different language, usually after every one of two lecture video there will be a execise for user to check if the had understand previous content of videos; moreover forum, quiz, assignments and other traditional xMOOC functions are also provide on Coursera. To make sure learners focusing on the lecture video, there is a pop-up question about current video in every lecturevideo see Figure 2.1. The question will let you submit three times, and if all the answers are wrong in three submission system will tell you the answer but learners can chose to continue the video without answering the question. Coursera also has iOS, Android, and Kindle Fire apps.

2.1.4 edX

edX [3] is also a xMOOC platform, which founded by Harard University and MIT at 2012 offering high-quality courses from universities and intuitions to learners. As a xMOOC too, edX and Coursera are pretty alike. Free for taking courses, center on lecture videos whit the supports like forum, exam, and other features most MOOC platform has. Similar to Coursera, edX has a certificate system, which you can apply for hard copy proof when you pass the course; it will charge you certificate fee, however, the price is slightly lower than Coursera. Besides the majority of courses are taught in english, edX also provides some foreign language courses.

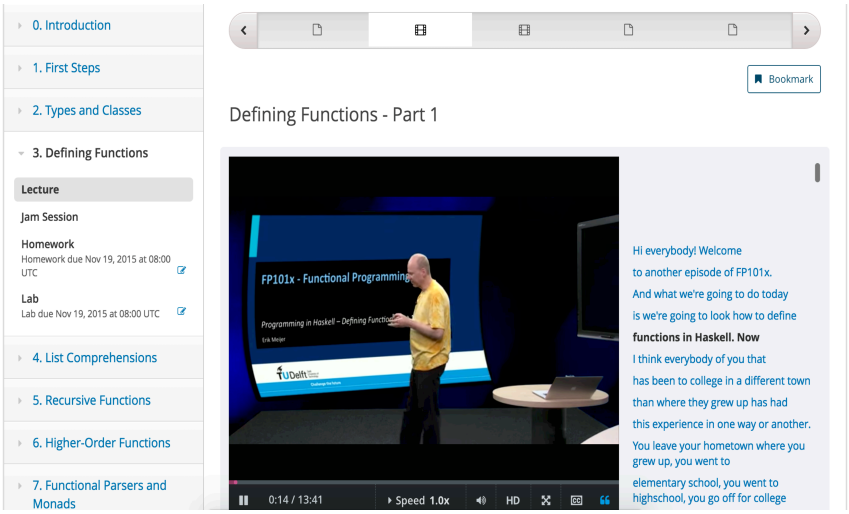


Figure 2.2: edX transcript redirect function: redirect video to specific section by transcript content

edX has transcript redirect function on the right of the player see Figure 2.2. This function is helpful for learners to know what is the video talks about before go through the whole video or help them easier to target the segment they what to replay again.

2.1.5 Open edX

Open edX is a free and open source course management system that was originally developed by edX. Among Open edX, there are many useful tool or model for constructing a course already be done, see Table 2.1. With the help of Open edX, people around the word can build and host their own MOOC, take xuetangx [10] for example, a famouse MOOC website in China for chinese speaking learners that built based on Open edX.

Feature	Description
Open edX Studio	Studio is the Open edX tool that used to build courses, which provided with a GUI interface make users easily edit courses through a browser. You use Studio to create the course structure and then edit course content, including problems, lecture videos, and other resources for learners. You also use Studio to manage the course schedule and the course team(instructors, tutors), set grading policies, publish each part of your course, and more.
LMS	LMS in the Open edX is a tool that learners use to access course resources and to check their learning progress in the course. The Open edX LMS can also offer a discussion forum and a wiki that both learners and course team members can contribute to.
XBlock	XBlock is the component architecture for the elements of an Open edX course. Platform developers can adjust course components to meet instructors' need. For instance, you can build XBlocks to represent course contents such as problems, text string, or HTML content to implement interactive learning laboratories. The primary advantage of XBlocks is that they are deployable in any edX Platform or other XBlock application, which make the code you write can be used by any course team and vice versa.
Open edX Insights	edX Indights is a course analytics system in Open edX, course team can use collected data to monitor learners' activities, learning patterns, reveal problems might confuse learners, or do researches base on these data to refine courses.

Table 2.1: Open edX features

2.2 Video Indexing Technology

2.2.1 Overview

hello

2.3 Data-Driven Analysis

2.3.1 Overview

Chapter 3

System Architecture

In this thesis, we design and implement a data-driven learning interest recommendation system and integrate with current MOOC platform. In this chapter, we will introduce the system architecture and course data flow in two different course scenario. Moreover, we will also introduce the website MVC design and the user activity modules.

3.1 Layer Structure Overview

This system can be divided into three parts: user interface, web server and data-service server. (see Figure 3.1). User interface is focus on precessing user's interaction and contains two parts, website view and data analysis view. Website view is contents peovided by web server, which contains all MOOC learning and management features for users to interact with, Data analysis view, on the other hand, is the content of analyzed result based on events triggered by users, which will presented by visualizing the processed data. Web server is mainly handling requests from user interface. Data-service is a API server collects user interface's activity datas and analyze it afterward, and furthermote send back the analyzed data re-

sult back to user interface whenever user send a specific request to data-service server.

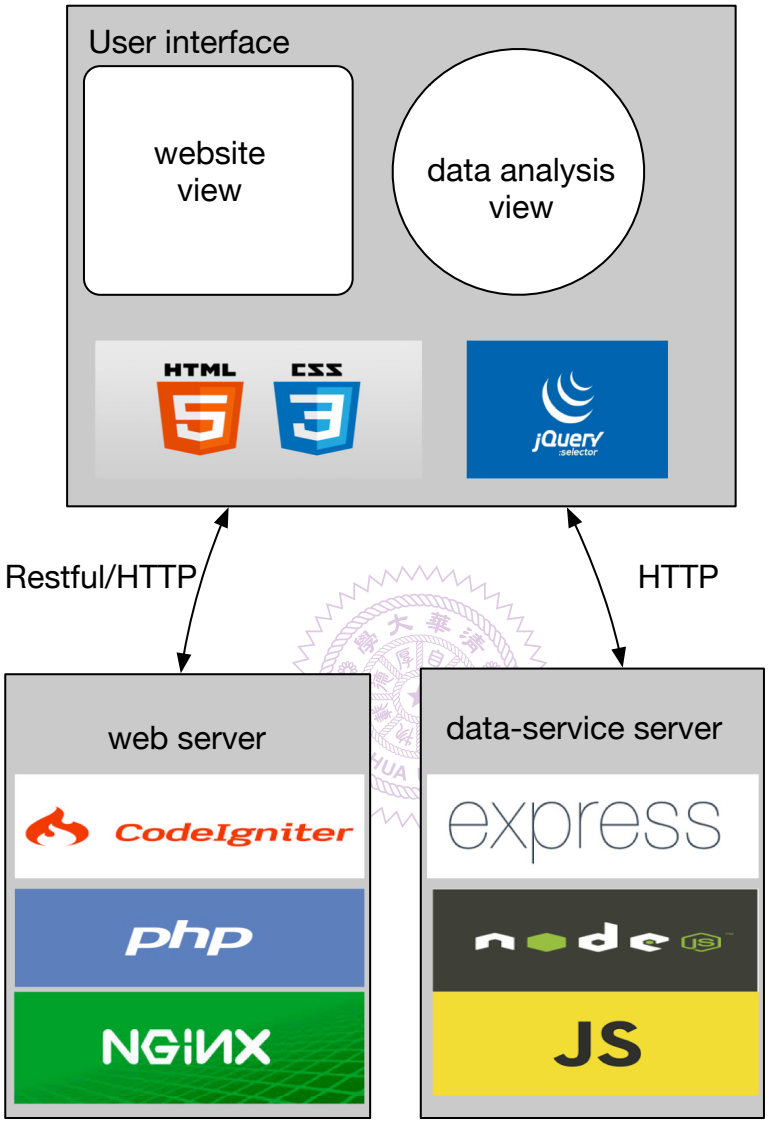


Figure 3.1: layer structure overview

3.2 Website MVC Design Pattern

MVC (Model-View-Controller) is a kind of design patter in software engineering, it seperate application into three interconnected parts, model, controller, and view [11] (see Figure 3.2). Controller act as the brain of application, dealing with users' request and converts requests into command for model and view. View provides user a Graphical User Interface (GUI) to display application's state. Model is the only one who can access data base directly, sometimes it represents a schema of table or specification or the data type.

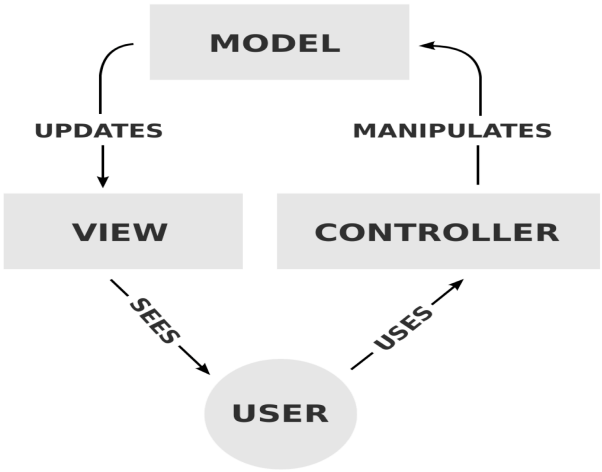


Figure 3.2: Collaboration of MVC elements

3.3 Activity Modules

User activity datas play an important role in the proposed system, in order to analyze user's activity pattern when learning on websit we designed four activity module for different scenario (see Figure 3.3) each activity module composed of multiple action which maps to user's activity. First, account module records user's

log into and logout from the website. Second, page module records user's trail since he/she log in to the website, every record is tagged with "leav" or "access" to note action that triggered the record so that we can monitor user's web page activity. Third, forum module has three action, "post", will records user's forum post, "reply" records which user give another postser a reply and "like" records user click like button on specific post and reply. Finally, video module is the most complecated module among activity modules, "play", "pause", "seek", "change rate" are actions that user interacts with player, and "leave when pause", "leave when play" are actions that user leave the lecture video page.

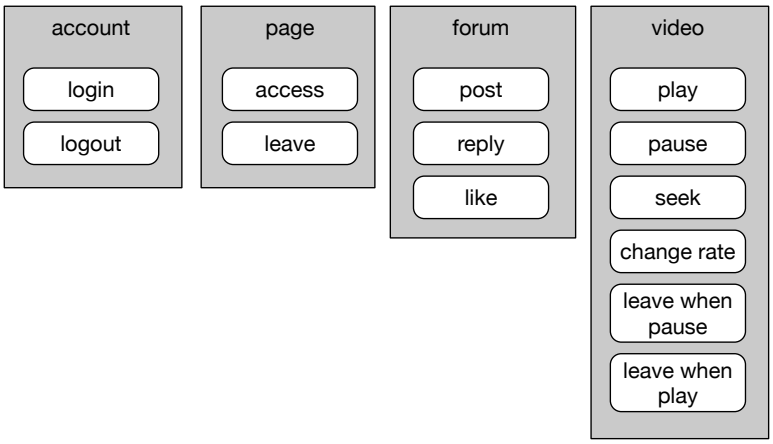
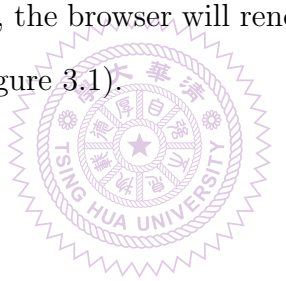


Figure 3.3: User activity modules

3.4 System Architecture

Figure 3.4 illustrates the system architecture, which shows the detail tn each parts. The whole system is constructured from two subsystem, web server and data-

service system, these two system are run independently. Web server is a entry for user to get access to this system, throug the browser user makes requests to web server to get pages they want; moreover it contains the most logic of web application and responsible for front-end presentation to the user. Data-service server is a pure API-server without front-end views, which responsible for collecting user activity datas and analyze collected datas. Whenever users interact with view element such as click buttom, click play on website embedded player, leave pages, etc. on the views provided by web servers (website view in Figure 3.1), a data will be sent to data-service server as a record. After every period of time, the data in data-service server will be analyzed by our algorithm and generate refined data. Finally, we send back the refined data back to user interface at the time users call the related api on data-service, the browser will render the result based on refined data (data analysis view in Figure 3.1).



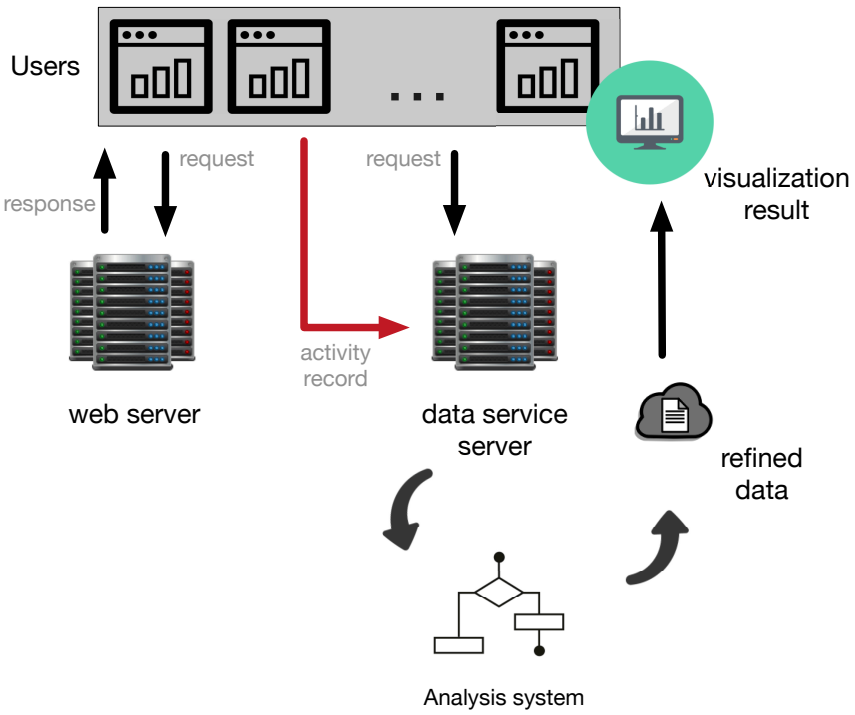


Figure 3.4: system architecture

3.5 Course Data Flow

On MOOCs, there are two course starting models can be seen, one is course that had open before and reopen, which is old course, the other is new course that never opened before. In view of keeping presented data updated, system should store refined data for user to access and simutaniously store new data for future analysis, to do so, we design two data flow for two different starting model. To begin with, data flow of new course (see Figure 3.5) focus on maintaining a schema for system easy to store and be scerialized. Moreover, to make system runs as a automation process we setup a worker to send a request to update datas to data-service server every period of time.

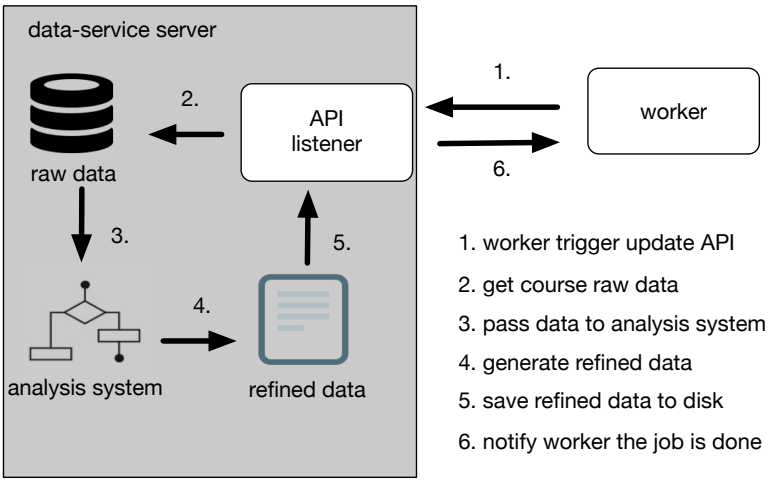


Figure 3.5: new course data work flow

For new course that had never opened before (see Figure 3.5), we simply query the data base and send the return data to analysis system and then generate and store back the new refined data to replace the old one.

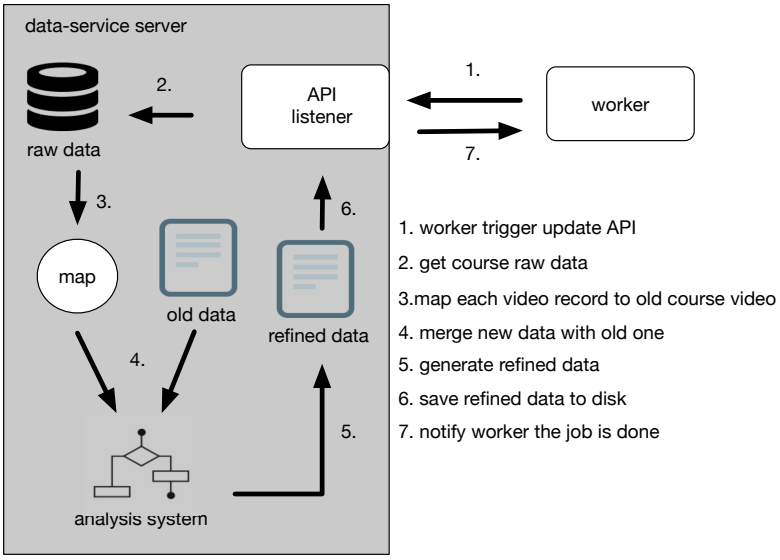


Figure 3.6: old course data work flow

On the other hand, processes to generate refined data for old courses is more complecated (see Figure 3.6), the challenge of the scenario is derive from data structure of MOOC platform, since there is no inherited relationship between courses. To solve this issue, we maintain a map table for classes that have inherited relationship to map each lecture video in the course, so that we can merge the old course data with new one.

Chapter 4

System Implementation

In this chapter, we will talk about the implementation datail of the proposed system and will focus on the function on data service server's environment and techeniques.

4.1 Data Service Server Environment

4.1.1 Node.js

Node.js [12] is a open sourse JavaScript application framework written in C [13], C++ [14] and JavaScript [15] Node.js can run on many operation systems like Microsoft Windows, Linux, Mac and embedded system like Raspberry, Cubieboard. There are many advantages of Node.js. To begin with, it use V8 JavaScript engine to boost performance, which make it much more better than other script language. Second, it's event driven and non-blocking I/O feature is surprisingly appropriate to both front-end and back-end of web application. Third, as a popular programming language like Python and Ruby, there are many third-party module on the word wild web that developers around the world wrote, we can make use of these modules to accelerate time developing apps. Whats more, the module is managed

properly by nodejs package management (npm), which make it easy to import modules.

4.1.2 Asynchronous Process Control

Since Node.js is JaveScript with extensional function, the event-driven feature in JaveScript is to Node.js, as a result we can often see asynchronous design model in JavaScript code. The idea of event-driven and asynchronous is run the job need long time to finish in the back-end event engine to avoid the job block all the process. After that, we will set a callback function to inform the process when the time-comsuming job is done. Figure 4.1 shows the idea of asynchronous process control.

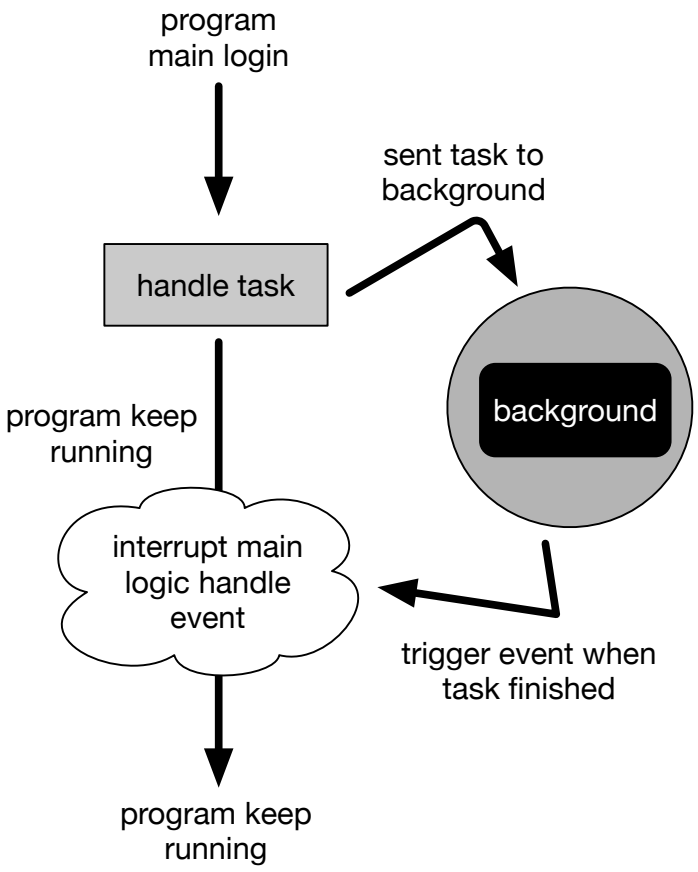


Figure 4.1: Asynchorous process control model

4.1.3 MongoDB

MongoDB [16] is an free and open source nosql database, data in mongodb saved as a json-like document with dynamic schemas (BSON). MongoDB can handle big data in scale in Terabytes with its high performance and it's tableless database structure also make it easy to integrate data in certain type of applications eaiser and faster.

4.1.4 Google Cloud Platform

Google Cloud Platform [17] is a platform integrates Iaas, Paas and Saas. According to the website, it provides eight categories of service, including basic virtual machine renting, database hosting, big data analysis, machine learning and more.

4.1.5 Iron Worker

Ironworker [18] is a service on Iron.io, which make developer setup a worker on website with flexible scheduling setting. Once developer setup the worker, Ironworker will monitor the worker, and if any thing go wrong, Ironworker will notify you by email with detail information.

4.2 Server Setup

In order to construct out server, we rent a VM instance machine type:n1-standard-4 (4 vCPUs, 15 GB memory), and on top of that we run a mongoDB on the same machine. After setting up the environment, we execute the nodejs api server conneted with the mongoDB to save the data to database and to provide other API for query data. Among APIs the server provide, there is a API will triggered by the worker we set onece a day, this API is designed for execute a python script, which is our data analysis system that will update the refine data (see Figure 4.2).

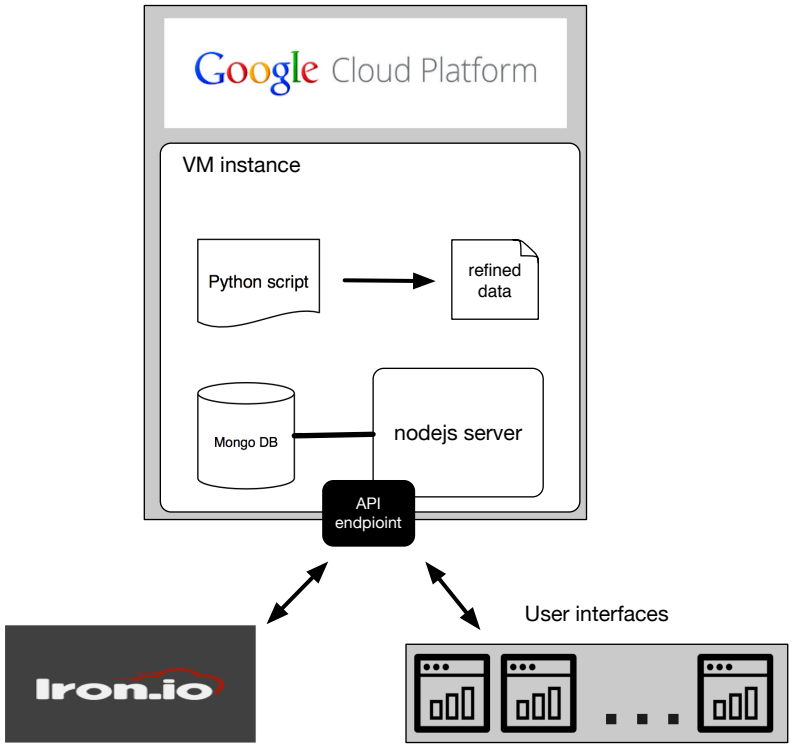


Figure 4.2: server setup

4.3 API Server

4.4 Analysis System

Bibliography

[1] Carol Yeager, Betty Hurley-Dasgupta, and Catherine A Bliss. cmoocs and global learning: An authentic alternative. *Journal of Asynchronous Learning Networks*, 17(2):133–147, 2013.

[2] Andrew Ng and Daphne Koller. Coursera. Retrieved May 15, 2016, from the World Wide Web: `https://zh-tw.coursera.org/`, 2012.

[3] Massachusetts Institute of Technology and Harvard University. edx. Retrieved May 15, 2016, from the World Wide Web: `https://www.edx.org/`, 2012.

[4] Mike Sokolsky Sebastian Thrun, David Stavens. Udacity. Retrieved May 15, 2016, from the World Wide Web: `https://www.udacity.com/`, 2012.

[5] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. Studying learning in the worldwide classroom: Research into edx's first mooc. *Research & Practice in Assessment*, 8, 2013.

[6] Daniel T Seaton, Yoav Bergner, Isaac Chuang, Piotr Mitros, and David E Pritchard. Who does what in a massive open online course? *Communications of the ACM*, 57(4):58–65, 2014.

[7] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 4017–4026. ACM, 2014.

[8] Akshay Agrawal, Jagadish Venkatraman, Shane Leonard, and Andreas Paepcke. Youedu: Addressing confusion in mooc discussion forums by recommending instructional video clips. 2015.

[9] Laura Pappano. The year of the mooc. *The New York Times*, 2(12):2012, 2012.

[10] Tsinghua University. Xuetangx. Retrieved June 2, 2016, from the World Wide Web: `http://www.xuetangx.com/`, 2013.

[11] A. Leff and J. T. Rayfield. Web-application development using the model/view/controller design pattern. In *Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International*, pages 118–127, 2001.

[12] Ryan Dahl. Node.js. Retrieved June 6, 2016, from the World Wide Web: `https://nodejs.org/en/`, 2009.

[13] Dennis Ritchie. C. Retrieved June 6, 2016, from the World Wide Web: `https://en.wikipedia.org/wiki/C_(programming_language)`, 1972.

[14] Bjarne Stroustrup. C++. Retrieved June 6, 2016, from the World Wide Web: `https://isocpp.org/`, 1995.

[15] Brendan Eich. Java script. Retrieved June 6, 2016, from the World Wide Web: https://developer.mozilla.org/zh-TW/docs/Web/JavaScript, 1995.

[16] MongoDB Inc. mongodb. Retrieved June 6, 2016, from the World Wide Web: https://www.mongodb.com/, 2009.

[17] Google Inc. Google cloud platform. Retrieved June 6, 2016, from the World Wide Web: https://cloud.google.com/, 2011.

[18] Iron.io. Ironworker. Retrieved June 6, 2016, from the World Wide Web: https://www.iron.io/platform/ironworker/.