清 华 大 学

综 合 论 文 训 练

题目：慕课学习过程数据采集、分析与展现

系　　别：计算机科学与技术系

专　　业：计算机科学与技术

姓　　名：罗富文

指导教师：向勇副教授

2016 年 6 月 16 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：_____ 导师签名：_____ 日　期：_____

中文摘要

随着 MOOC(大规模在线课程) 课程平台的普及，高校的课程可以通过网络让成千上万的学生同时学习，但是这也为老师带来了新的问题，无法及时了解每个学生的状态。因此一个可以收集并跟踪学生学习状态的系统对 MOOC 平台的教学活动就显得十分有意义。

本文分析了 Open edX 现有的对学习分析功能的支持情况，同时介绍我开发的学习分析系统。这个系统采用可扩展的结构，通过收集，分析，可视化等一系列步骤完成对多个平台的学生数据的收集与整合工作。我们使用这个系统收集了 Open edX，GitLab 以及 Piazza 平台上的学生数据。系统支持多种可视化方案，可以对不同学生的数据进行对比查看，同时，可以自动为每个学生和整个班级生成一份综合报告。

这个学习分析系统的创新点主要有：

- 完全独立于 Open edX 系统
- 支持从多平台收集并整合学生数据

关键词：慕课；Open edX；学习分析；多平台数据收集；可视化

ABSTRACT

With the development of MOOC(Massive Open Online Courses) platform, thousands of students can learn courses online. It's necessary to build a powerful learning analytics system which can collect and track status of students since teacheres can no longer be aware of so many students.

This thesis first discusses the learning analytics suport on Open edX, then introduce our work, the learning analytics system working on our Operation System course. We collect student data on Open edX, Piazza and Gitlab, then visualiaze the result with our visualization module. In addition, we can generate reports for each student and whole class.

There are some points making our system different:

- Independent from Open edX platform
- Collect data from multi-platforms

Keywords: MOOC; Open edX; Learning Analysis; multi-platform data collecting; visualization

目 录

第 1 章 引言 1
1.1 背景 1
1.2 相关工作 2
1.2.1 Open edX 平台 2
1.2.2 基于 Open edX 的操作系统课程平台 2
1.2.3 其他平台 3
1.2.4 可视化方案 4
1.2.5 总结 4
第 2 章 学习分析系统 6
第 3 章 数据收集 9
3.1 事件数据格式 9
3.2 操作系统课程平台的数据收集 11
3.3 Gitlab 平台的数据收集 16
3.3.1 Gitlab 提交记录的收集 17
3.3.2 在线实验成绩收集 18
3.4 Piazza 平台的数据收集 19
3.5 总结 20
第 4 章 数据分析 21
4.1 数据分发 21
4.2 数据处理 21
4.2.1 消费者 21
4.2.2 消费者管理模块 22
4.2.3 数据处理结果 24
4.2.4 结果池 25

第 5 章　数据可视化 27
5.1　可视化模块 27
5.2　可视化结果示例 30
5.3　学生和班级的综合报告 32
第 6 章　总结 35
插图索引 36
表格索引 38
参考文献 39
致　谢 40
声　明 41
附录 A　外文资料书面翻译 42
A.1 引言 42
A.2 相关工作 43
A.3 edX 上的学习分析 44
A.4 技术和结构概述 46
A.5 可视化实例 50
A.6 结论和未来工作 52

第 1 章　引言

1.1　背景

MOOC(大规模在线课程) 是近几年兴起的一种在线课堂，通过网络，来自不同地区的学生可以参与到同一门课程中。目前比较重要的 MOOC 平台有 Coursera，edX，以及 Udacity，而国内也逐渐产生了很多优秀的的 MOOC 平台（例如学堂在线，中国大学 MOOC），国内也有越来越多的 MOOC 学生用户，这也随之带来了一些问题。MOOC 平台上的学生数量庞大，而且很可能来自不同地方，因此对于老师来说要了解整个班级的学习情况就变成一个基本不可能完成的事情。然而，了解整个班级的学习情况以及每个学生的个人情况是教学工作中一个非常重要的模块。所以为 MOOC 平台提供整个课程的统计结果就变得十分有意义。

学习分析则是符合这一需求的一个最近逐渐兴起的领域。学习分析的思想就是通过收集和统计学生在学习过程中的行为等数据，以了解课程下学生的整体情况（例如成绩分布，学习课程投入的时间等）。这些结果可以帮助课程管理人员很好地改进自己的课程，从而改善用户的课程体验。

为了进一步推广 MOOC 课程，edX 已经开源为 Open edX，其对应的源代码可以在 Github 上找到。Open edX 的文档丰富，同时 Open edX 的开源吸引了一大批优秀的开发者，带来了活跃的社区，这些让 MOOC 课程平台的开发变得简单。

清华大学计算机科学与技术系开设的操作系统课程是清华大学使用 MOOC 平台的课程之一，操作系统课程借助 Open edX 搭建了自己的在线课程平台。操作系统是一门着重实践的课程，因此除了理论学习，课程还引入了在线实验来帮助同学更好的了解操作系统。同时，为了满足同学们的问答需要，操作系统引入了 Piazza，一个第三方的课程讨论平台，以方便学生在遇到问题的时候可以相互讨论或者请教老师。

我们的工作则是基于清华大学计算机系的操作系统课程的在线平台，为其开发涉及多个平台的数据收集，统计与分析系统。

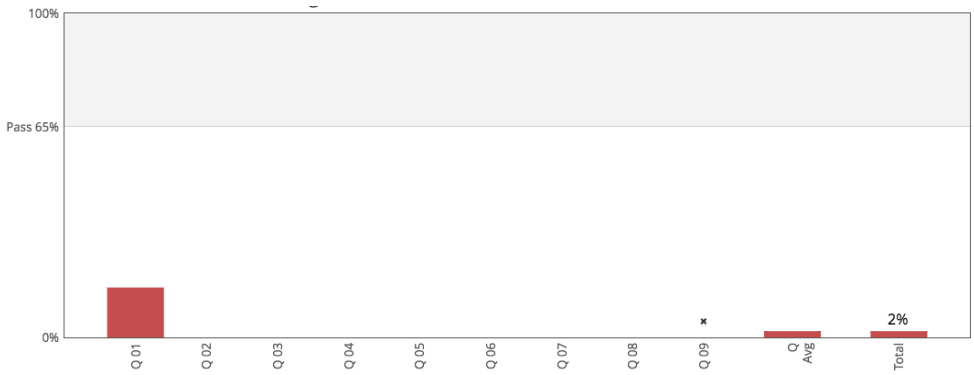

图 1.1　Open edX 上的学生个人作业完成情况

1.2　相关工作

与一般 MOOC 课程不同，我们的操作系统课程涉及到较多的平台，除了最基本的 Open edX 课程，还涉及到 Piazza 平台，同时管理学生的实验代码需要用到到 gitlab 平台和 github 平台。这一节我们先分别介绍这几平台对学习分析的支持，然后我们再讨论我们工作的可行性以及可能会遇到的困难。

1.2.1　Open edX 平台

Open edX 是由 edX 开源出来的平台[1]，内置了一些支持学习分析的功能，例如学生作业的完成情况的统计（见图 1.1），也有很多开发者为 Open edX 的学习分析提供了很多方案 [2]。文 [3] 基于 Open edX 架构，提出了自己的学习分析扩展，可以统计和收集学生观看在线视频的时间情况和课程整体成绩分布情况。

因为我们的操作系统课程的在线平台也基于 Open edX，所以我们的数据收集工作也主要基于 Open edX。Open edX 将课程的内容细分为一个一个的模块 [4]，在项目的内部，这些模块被称为 XBlock。XBlock 概念的引入让 Open edX 的管理和维护变得很容易，同时 Open edX 官方也提供了丰富详细的文档说明，这为我们工作涉及到的 Open edX 提供了便利。

1.2.2　基于 Open edX 的操作系统课程平台

我们的操作系统课程虽然涉及到很多平台，但是 Open edX 是其中最核心的平台。这个平台是学生进行在线联系的平台，同时也是在线实验，课程 wiki，piazza 讨论区的主要入口。操作系统课程平台基于 Open edX 的 XBlock 的概念

进行较多的开发。

在线练习的 XBlock 是其中比较重要的一个 XBlock，原生的 Open edX 已经有对在线练习的支持，但是因为原生的在线练习没有提供对题库的支持，每次布置练习老师和助教都需要手动输入题目内容和题目答案，这样不仅增加了工作量，而且布置题目的出错率也很高。有了操作系统课程平台的这个扩展，老师和助教只要通过制定题目编号就可以完成题目的布置。同时为了可以进一步分析学生们的回答情况，操作系统课程平台将学生在线练习的统计结果保存到 gitlab 上。

操作系统课程平台中还有其他第三方扩展，例如 piazza 讨论 XBlock，让学生可以不登录 piazza 直接从操作系统课程平台中查阅 piazza 中的问题；代码浏览平台和实验 docker 环境初始化平台，让学生可以自己完成实验环境的初始化工作和代码的浏览工作。而为了更好的与原有的架构兼容，我们的学习分析系统也会采用 XBlock 的概念，通过开发第三方的 XBlock 为我们系统的处理结果提供可视化结果的展现方法。我们将在后面的章节详细介绍这些工作。

1.2.3 其他平台

除了操作系统课程平台，我们的工作还涉及到 Piazza 平台以及 gitlab 和 github。Piazza 平台是一个相比 Open edX 的讨论区功能更加全面的平台，提供了很多额外的功能，同时 Piazza 平台会自动统计和处理学生的问答数据，我们可以通过对应的 API 调用来获取 Piazza 的统计结果。操作系统基于 gitlab 为学生搭建了在线实验代码管理平台，为了方便起见，在本文中我们以 gitlab 来简称对应的代码管理平台。学生通过将实验代码提交至 gitlab 平台，系统就可以很方便的获取学生的代码并进行批改。github 是一个第三方平台[①]，它提供与 gitlab 一样的功能，我们的在线实验允许学生从 gitlab 或者 github 中选择一个作为自己的代码管理平台。

涉及多平台也为我们的工作带来了一些问题，一个最主要的问题就是我们很难去确认每个平台上的账号之间的关系，也就是说，对每个学生在操作系统课程平台，Piazza，gitlab 或者 github 上都有一个账号，而这些账号之间是没有明显的关联的。对于我们的学习分析系统而言，确认每个学生的账号关联就变成一个非常麻烦的工作。还有一个问题，我们的系统需要同时收集各个平台的

① https://github.com

数据，但是这些平台的数据格式以及获取数据的方式都个不相同，需要获取数据的频率也不尽相同，这无疑会给我们的工作带来额外的工作量。

1.2.4 可视化方案

对处理的结果进行可视化是学习分析中非常重要的一个模块，通过对处理结果使用可视化方法，老师和学生可以更直观的了解处理结果背后的意义。在工作的初期调研阶段，我们对现有的可视化库进行了调研，主要围绕其支持的图表数量，开源情况，以及交互性和社区支持等情况进行了调研。主要的结果可以参照表 1.1。

表 1.1 可视化库的调研结果

可视化库	支持的图表数量	开源	交互性	对移动端的支持程度	社区支持
d3.js	超过 100 种	是	一般	一般	较好
Charts.js	6 种	是	较好	一般	较好
XCharts	2 种	是	一般	一般	一般
Aristochart	仅支持折线图	是	较好	一般	较好
HighCharts.js	约 15 种	否	较好	较好	较好

D3.js 是目前最流行的可视化库之一。但是它主要提供的是绘图方面的可视化支持，也就是说，D3.js 不直接支持对图表的绘制，它提供的是绘制图表的基础操作函数，例如绘制曲线，三角形，扇形的函数。Charts.js 和 XCharts.js 则是基于 d3.js 开发的可视化库，其调用 d3.js 提供的绘图方法并包装成提供显示图标的工具库。Aristochart 也是基于 d3.js 的可视化库，其专注于折线图的显示，不支持其他类型的图标。HighCharts.js 是相对于 d3.js 更高层的库，提供多种图标的可视化方法，同时 HighCharts.js 有良好的交互性，用户可以自定义很多的交互事件，值得一提的是 HighCharts.js 不是开源的项目，但是对于非商业用途是完全免费的。HighCharts.js 提供完善的可视化方案，以及丰富的社区支持。

1.2.5 总结

考虑到操作系统课程对多平台的以来性，我们最终决定让我们的学习分析系统支持多平台，同时希望将系统的架构进行一定的设计，使得系统可以轻松的对收集数据的源进行修改和扩展。同时，考虑到需要收集的数据的规模，我们

需要系统可以支持增量而不仅是全量的收集以减少每次收集的代价。同时，因为需要对多个平台的数据进行收集，分析和统计，为了让老师更方便的理解分析结果，我们需要为结果提供一定的可视化方案。考虑到我们系统的需求，我最终决定使用 HighCharts.js 作为我们的可视化工具库。

本文将按照以下的顺序介绍我们的工作：我们会先在第 2章对我们的学习分析系统进行总体的介绍；在第 3章，我们会介绍在各个平台上的收集工作，同时介绍我对原有系统的一些改进；在第 4章，我们会依次介绍系统的数据分发和处理过程，并介绍我们能够提供的分析功能；在第 5章，我会介绍数据可视化模块的主要结构以及我们可以提供的可视化方法。最后，我们在第 6章对我们的工作进行总结，同时讨论未来可能的工作。

第 2 章　学习分析系统

我们的学习分析系统的架构可以参照图 2.1，整个数据收集和分析系统主要分为三个模块：收集模块，处理模块和可视化模块。其中收集模块主要由收集者组成，处理模块由分发者和消费者组成。

收集者 (Collector) 的主要功能是收集数据，考虑到我们要从多个平台收集数据，因此我们把收集的过程分配到独立的模块，针对每个数据源制定不同的收集方案。**分发者** (Dispatcher) 负责接收收集者发出的数据，并根据数据的信息将数据分发给下一级也就是消费者。**消费者** (Consumer) 是我们整个系统进行数据分析的主要角色，它向分发者订阅自己需要的数据，并对数据进行处理。

除了三个主要模块，我们系统还有两个数据池，事件池 (Event Pool) 和结果

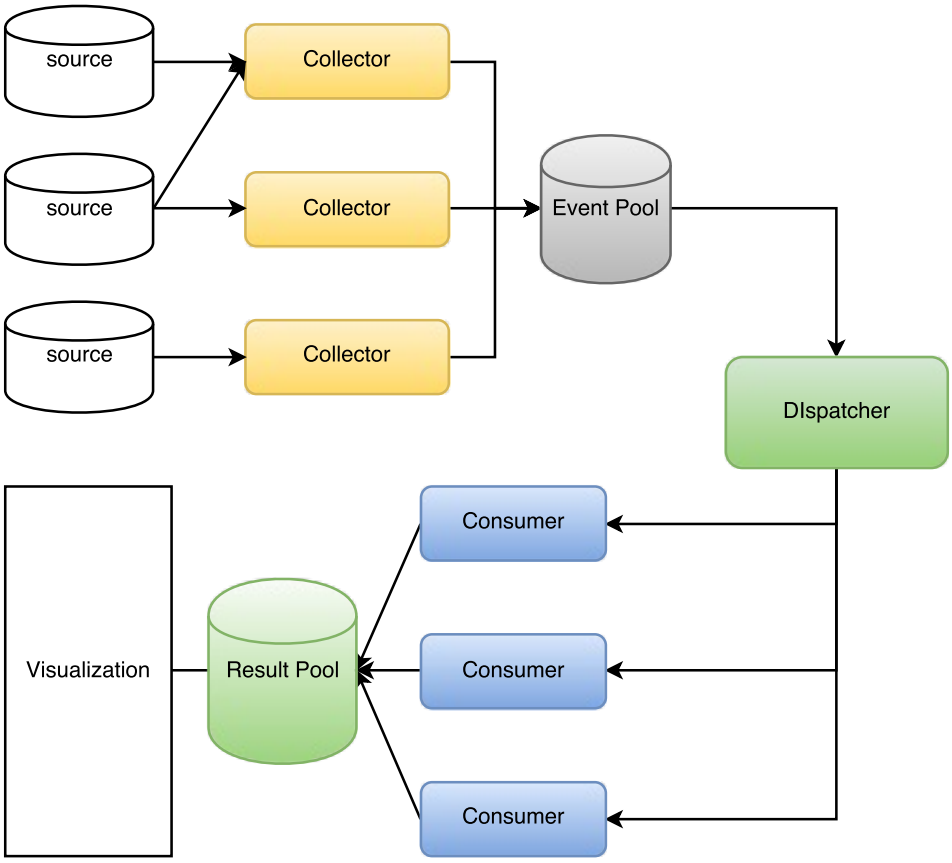

图 2.1　学习分析系统主要架构

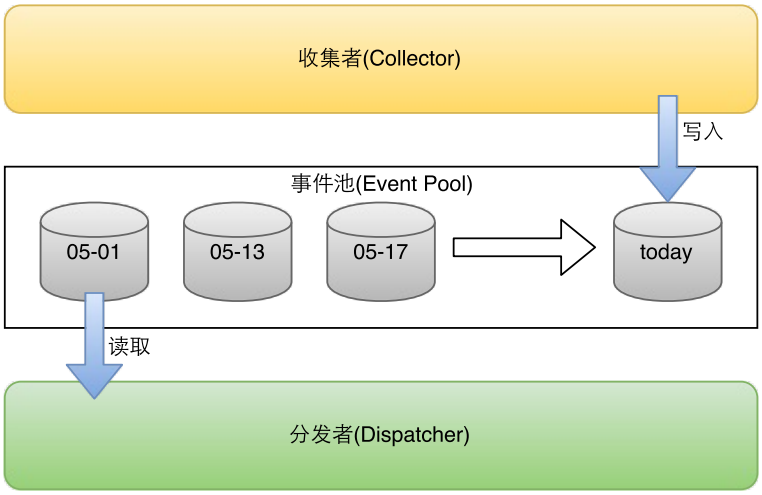

图 2.2　数据收集和分发

池 (Result Pool)。两个事件池在三个模块中担任数据交互和缓冲的角色。

增量处理为了满足增量处理的需求，我们的数据收集系统将和学生相关的每个行为抽象为一个事件数据，收集者以事件为单位收集数据并将数据写入事件池，例如，某个学生在操作系统课程平台上回答了某个问题，我们的收集者就会产生对应的事件数据，并将事件数据发送到数据池中；同时，当这个学生在这一题的回答被批改时，收集者又会产生另一个事件数据。通过追踪一系列的事件来跟踪并收集每个学生的状态，从而完成数据的收集。所有收集的事件数据在事件池中以收集的时间进行排序，这样做的好处是，我们可以以事件为单位对处理结果做增量修改，同时可以容易的将处理结果进行回滚，例如，如果我们希望系统数据回滚到 5 月 1 日时的状态，只需要过滤掉 5 月 1 日以后的数据，然后对剩余的数据进行一次全量处理即可。

如何追踪事件？我们的数据收集涉及到多个相互独立的平台，每个平台都有自己的用户管理系统，因此将每个学生在多个平台上的数据进行收集并整合的最大的困难就是如何确定学生在不同平台的账户之间的联系。在这里，我们让学生通过操作系统课程平台的在线练习系统维护自己在其他平台的信息，然后由我们的系统将这些信息进行收集和保存，最后在根据这些信息来确定和数据相关联的学生。

分发者定时从事件池中获取事件数据，并根据事件的类型和主题将其分发给消费者，消费者完成数据的分析工作后将结果以固定的格式写入结果池中。我们会在第 4章继续讨论。

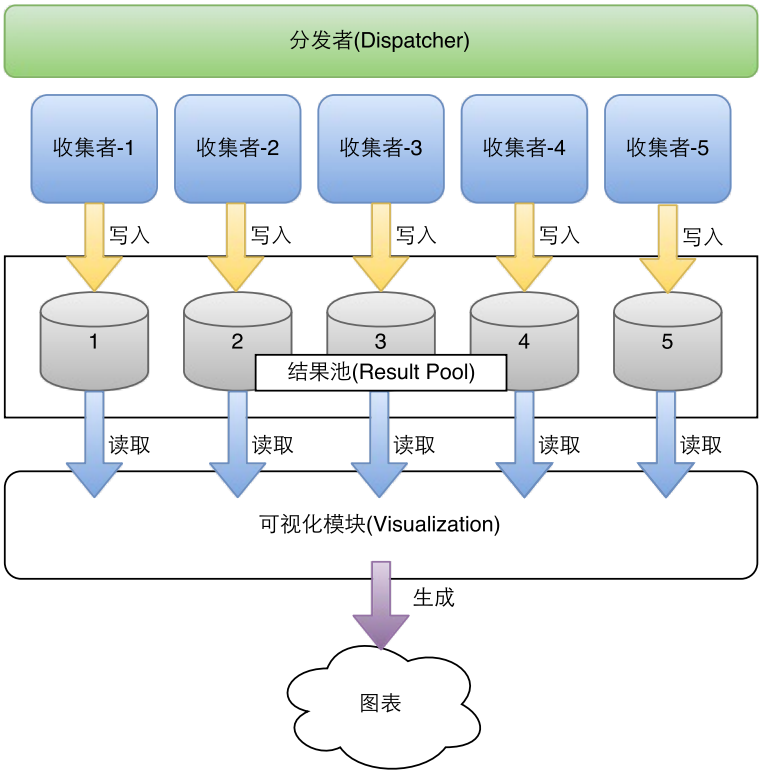

图 2.3　数据分析和可视化

结果池中的数据就是我们学习分析的结果了，但是这并不完全是用户需要的数据，虽然处理结果相比于原始数据已经精简很多，但是对于老师来说，理解这些数字仍然有很大的困难。系统中的可视化模块负责对学习分析的结果提供可视化支持，让处理结果变成可以容易理解的图表。有时候老师还需要横向比较各种结果，例如将某个学生的每一章节在线练习的成绩和全班的平均成绩比较，因此可视化模块还要支持数据横向比较。我们会在后面的章节详细介绍这个模块。

第 3 章　数据收集

收集者是我们的数据收集模块的主要角色，学生在操作系统课程平台平台以及其相关平台上进行的大部分操作都会产生对应的数据，因此我们将每次产生的数据都抽象为一个个事件，通过收集每个事件，收集者也就完成了对数据的收集。收集者是独立于我们的学习分析系统的。这主要体现在以下几点：第一，收集者和分发者只通过事件池进行交互。收集者向事件池中写入事件数据，分发者从事件池中读取事件数据，因此收集者的写入工作和分发者的读取工作是相互独立的，即使收集者发生改变 (例如种类和数量的改变)，分发者也不会受到影响，在这种情况下，收集者和分发并不要求使用相同的编程语言，甚至不需要运行在同一台机器上。第二点，不同的收集者之间也是相互独立的。每个收集者只需要关心其收集的数据源，不同的收集者之间没有直接的联系，因此只要收集者写入事件池的事件数据符合要求，每个收集者的实现方式和编程语言都可以自由指定。

每一个平台都可能会有多个数据收集来源，例如，对于操作系统课程平台来说，学生答题结果和题目的批改结果就是不同的数据源，一般来说，每一个数据源会对应一个收集者，当然收集者根据实际情况也可以同时收集多个数据源的数据。

3.1　事件数据格式

事件是我们处理数据的最小单元，学生在不同平台的交互行为会产生不同类型的事件数据，而每条事件代表的信息很可能会完全不同，例如来自在线平台事件数据会涉及到学生信息，题目信息，答案信息和成绩信息等，而来自 Piazza 平台的事件数据则可能会涉及学生信息，以及提问信息等。为了尽可能兼容所有平台的事件数据，同时考虑到扩展性，我们使用 json 格式的数据来储存每一个事件。虽然 json 是一个可扩展的字段，但是为了方便处理，我们还是对事件 json 一些字段做出要求。表 3.1是我们 json 数据中的部分字段的说明。

事件数据中的一部分字段是由事件决定的，例如 content 字段，该字段可以用来根据不同的平台的需求保存不同数据。event_type 字段主要用于指定事件涉及

表 3.1　事件 Json 数据的主要字段说明

字段名	类型	描述
event_type	string	事件类型
related	json object	事件相关信息（例如学生信息）
topic	string	事件的主题
time	utc string	事件发生的时间
collector	string	事件对应的收集者
content	json object	事件的主要内容

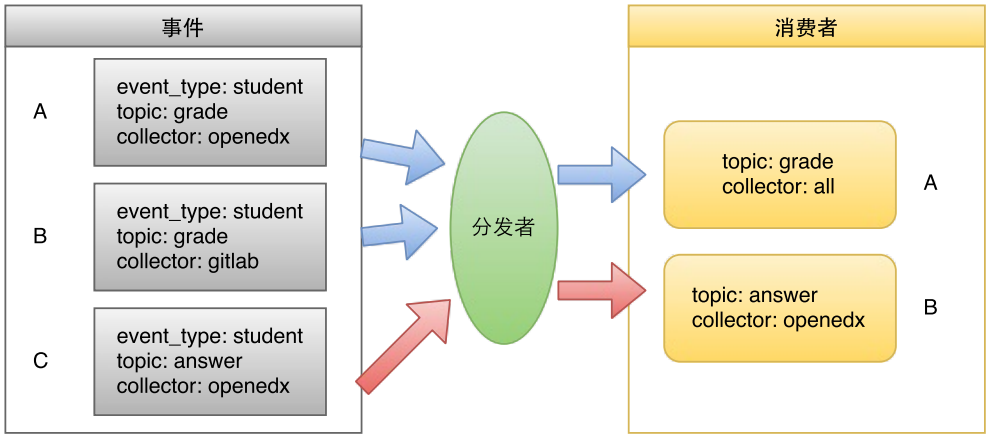

图 3.1　事件的分发过程

到的主要对象，大多数情况下，我们收集的事件的主要对象都是学生，但有时候也会遇到主要对象是课程的事件，例如课程的开始和结束事件。通过对事件 json 数据引入固定的字段，我们后期的事件处理和分发工作可以得到简化。因为分发者可以根据事件的这些字段来决定要把事件分发给哪些消费者。在图 3.1的例子中，事件 A 是学生的在线练习批改事件，因此其 event_type 字段指定为 student，topic 字段指定为 grade，代表这则事件主题是评分，collector 字段为 openedx，表示这则事件数据的来源是 openedx 平台也就是我们在线练习平台的收集者。事件数据通过这几项数据来告诉分发者自己的信息。同样，在消费者端也通过类似的字段来选择需要的事件数据，在示例中，事件 A 和事件 B 最终会分发给收集者 A，而事件 C 分发给收集 B。

图 3.1没有表现出来的一个事件数据的特征是，同一个数据可以被多个消费者订阅，在这种情况下，分发者会同时将一则事件分别发送给不同的消费者。

```
{
    "content":{
        "answer":"AB",
        "desc":"student({student}) answer question({question}) in course({course})"
    },
    "event_type":"student",
    "related":{
        "course":"openedx-os",
        "question":{
            "type":"multi_answer",
            "q_number":1310
        },
        "student":{
            "username":"stu",
            "email":"test@mail.com"
        }
    },
    "topic":"answer",
    "time":"2016-03-09:09:24:24",
    "collector":"answer-collector"
}
```


图 3.2　事件 json 数据样例

图 3.2给出了学生在操作系统课程平台回答问题产生的事件数据的一个样例。样例中 event_type，topic 和 collector 字段已经在之前介绍，time 字段代表的是事件发生的事件，大多数时候一个事件发生的事件和事件被收集的时间不一样，因为收集者是定时运行收集数据，因此数据会有一定的滞后。content 字段是由事件自己定义的，在示例中，content 里定义了 answer 字段以及 desc 字段，分别用来保存学生的答案以及对应的描述。还有一个值得注意的是 related 字段，这个字段保存了事件涉及到的相关信息，示例事件设计到操作系统课程，在线练习的题号，学生本人，这些信息分别以'course','question','studetn' 字段显示。通过这些相关信息，我们的系统可以跟踪一系列的相关事件，从而了解每个学生的状态。

3.2　操作系统课程平台的数据收集

操作系统课程平台是整个操作系统课程线上运行的核心部分，因此我们大多数的数据收集工作都会围绕操作系统课程平台来展开。正如我们在 1.2.1节已经讨论过的，操作系统课程平台原有的开发是基于 Open edX 的 XBlock 概念的。因此我们在操作系统课程平台上的收集工作也会围绕 XBlock 展开。

然而，在操作系统课程平台进行数据收集工作时，我们也遇到了新的问题。

```
2
lab6的调度过程包括() s2
- [x] A.触发: trigger scheduling
- [x] B.入队: 'enqueue'
- [x] C.选取: pick up
- [x] D.出队: 'dequeue'
- [x] E.切换: process switch

> 知识点: 进程和线程管理。
> 出处: 网络
> 难度: 1
> ABCDE
```

图 3.3　原始的 markdown 格式的题目样例

原有的在线回答系统，只是一个很简略的版本，在学生答题时，并没有收集到足够的数据，这为我们的数据收集工作增加了难度，同时结合学生对原有答题系统的使用反馈和意见，我们决定对操作系统课程平台原有的答题系统进行重新设计和开发，以满足我们的需求。

操作系统课程平台的答题系统包括题库系统和问答系统。题库系统是由老师和助教使用的题目编辑系统，原始的题目保存为一个个 markdown 格式的文件，这些文件保存在 github 上。这样的方式带来了很多问题。第一，markdown 格式的题目数据很难保存所有的信息。例如题目来源，题目难度，题目答案等。让计算机从 markdown 文本中分辨出题目，答案，难度，来源等信息是比较困难，同时因为，题目的 Markdown 文本由老师们人工编辑，难免会出现语法错误或者信息遗漏，这增加了题目维护的出错率。第二，因为题目数据保存在 github 上，老师只能通过第三方的方式来维护题目，这增加了题库维护的工作量。

考虑到 markdown 带来的种种问题，我们决定使用 json 格式的字符串来储存每个题目的信息。图 3.3和图 3.4分别是原始的 markdown 和升级后 json 格式的样例。我们可以发现升级后的格式在数据管理上更加清晰。借助 json 格式的优点，我们可以很容易对题目数据进行管理，可以很好的解决 markdown 不能保存所有信息的问题。同时，json 清晰的格式为我们后续的题目处理可以提供更好便利。

为了改进对 json 格式数据的维护过程，我们在操作系统课程平台上开发了对应的 XBlock 扩展。借助这个扩展，老师和助教就可以直接在操作系统课程平台平台上进行题库的维护工作。

有了 json 格式的题目数据，我们就可以对问答系统进一步进行改进了。这

```
{
    "status":"ok",
    "knowledge":[
        "进程和线程管理"
    ],
    "degree_of_difficulty":1,
    "explain":"ABCDE\n",
    "question":"lab6的调度过程包括() s2\n",
    "source":"网络",
    "answer":"ABCDE",
    "type":"multi_answer",
    "options":[
        "A.触发: trigger scheduling",
        "B.入队: 'enqueue'",
        "C.选取: pick up",
        "D.出队: 'dequeue'",
        "E.切换: process switch"
    ],
    "q_number":1442
}
```


图 3.4　升级后的 json 格式的题目样例

图 3.5　操作系统课程平台上的题库编辑 xblock

图 3.6　升级后的问答系统 XBlock

也是我们在操作系统课程平台信息收集的核心步骤。为了兼容 Open edX 原有的风格，我们将新的问答系统的外观设计为与原始版本相似。

为了尽可能还原学生答题的状态，对于学生的每一次回答，我们需要收集以下信息：

- 学生信息
- 题目信息
- 学生进行回答的时间，答案的内容以及历史回答

除此之外，为了兼容 Open edX 的对于学生答题次数的限制[①]，我们也增加了对学生当前答题次数的统计。图 3.7是学生进行某次回答产生的数据。

值得注意的是我们在问答系统产生的数据中提供了问题的快照，见图 3.7表示的 json 数据的 question 字段。对于回答问题的数据记录，其实我们只需要记录回答问题的具体题号就可以根据题号获取到题目的所有信息，但是我们这里选择了保存整个题目的信息。这是因为老师和助教可以通过题库系统随时修改任意个题目，包括题目的描述和题目的答案。因此可能发生这样的事情：学生在回答题目时，选择了正确答案，这里我们假设正确答案是 A，但是在这一题被批改之前，老师对这一题进行了修改，将这一题改成了完全不同的新的题目。这个时候因为题目的题号没有发生变化，学生的回答仍然有效，但是其答案本身已经没有意义。为了尽量保存学生回答的现场信息，每次学生的回答我们都会同时保存当时题目的快照，这样做是为了避免对题目的修改造成的各种可能

① Open edX 的默认功能，允许同一学生对不同的题目进行多次尝试回答

```
{
    "answer": [
        {
            "answer": "B",
            "time": "2016-04-09:18:19:19"
        }
    ],
    "maxTry": 0,
    "question": {
        "status": "ok",
        "knowledge": [
            "调查问卷"
        ],
        "degree_of_difficulty": 1,
        "explain": "解释: 统计学生的年龄组成情况。\n",
        "question": "年龄\n",
        "source": "网络",
        "answer": "B",
        "type": "single_answer",
        "options": [
            "A.19岁以下",
            "B.20~29岁",
            "C.30~39岁",
            "D.40岁以上"
        ],
        "q_number": 1132
    },
    "tried": 1,
    "student": {
        "username": "Winton1881",
        "email": "luofuwen5935@sina.cn"
    }
}
```


图 3.7　回答数据格式样例

的数据错误。

到这里我们基本上完成操作系统课程平台的信息产生工作。我们通过完善原有的平台，来收集之前不能收集的信息。操作系统课程平台产生的数据都保存在 gitlab 上，对于操作系统课程平台来说，gitlab 扮演了数据库的角色。所有的和在线练习相关的数据，包括学生回答数据和系统批改数据，都保存在 gitlab 上对应的库中。因此通过 git 命令提供的对应方法，我们可以很容易的跟踪数据的变化。

如何处理增量更新 我们的收集者可以理解为一个定时启动的程序，每次启动都会按照图 3.8流程进行工作。每次启动时，收集者会让本地库与 gitlab 中的对应库进行同步。然后根据 git 提供的命令比较最新的 Commit[①]和上一次 Commit，获取两次 Commit 之间发生改变的文件。再通过检查这些文件列表，我们就可以

① git 对每次提交都会产生对应的 Commit ID，这些 ID 是各不相同的

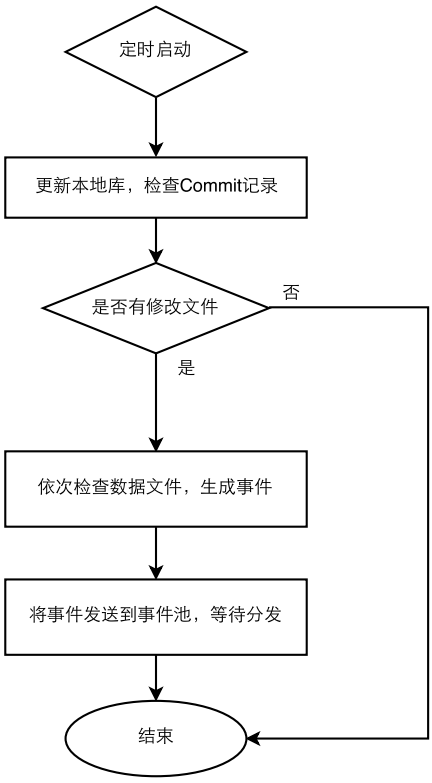

图 3.8　操作系统课程平台收集者工作流程图

知道从上一次统计以后有哪些新的数据产生。根据这些变化，我们就可以让收集者生成对应的事件。同时将事件数据发送到事件池，等待进一步处理。

3.3　Gitlab 平台的数据收集

在操作系统课程中 Gitlab 主要负责保存和批改学生的在线实验的代码，我们可以通过收集学生与 Gitlab 平台的交互行为来了解学生在在线实验方面的状态。

Gitlab 是第三方平台，与操作系统课程平台不同的是，Gitlab 内置了大量的数据统计功能，同时已经给出了足够清晰的 API 以及相关的文档[5]。Gitlab 是与操作系统课程平台相互协作的平台之一，其主要功能是保存学生在线实验的代码。我们主要收集以下信息：

- 学生的实验提交记录
- 在线实验成绩

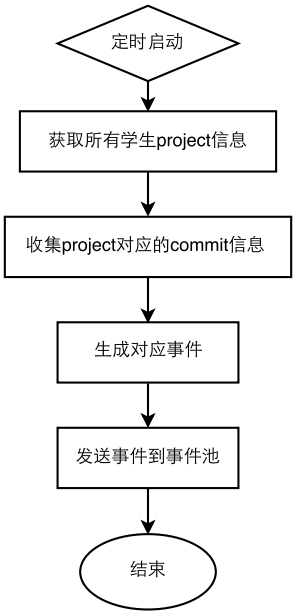

图 3.9　gitlab commit 收集者工作流程图

3.3.1　Gitlab 提交记录的收集

在章节 1.2.3中我们已经提到在线实验允许学生从 gitlab 和 github 平台选择一个平台作为自己的实验代码管理平台，不过因为 github 平台是第三方的代码管理平台，在收集数据方面我们会面临各种各样的权限问题，因此我们目前只能收集 gitlab 上的信息。

我们通过图 3.9所示的流程收集学生在 gitlab 平台生的 commit 数据。首先收集者会使用教师账户登录 Gitlab，调用 API 获取所有代码库的信息，包括库的所有者信息。对于每个代码库，判断其是不是当前选课学生的代码库，如果是，则通过 API 获取学生在这个库的 Commit 记录并生成对应的提交记录的事件，最后将生成的事件发送到事件池。

一个需要注意的问题是，gitlab 虽然提供了获取最近提交记录的 API，但是只能通过简单的参数对结果进行筛选，例如，获取最近 n 条提交记录。这个 API 无法支持更高级的调用比如获取从某次提交以后的所有提交记录。因此数据的筛选工作只能在收集者内部完成。一个可能的实现方法是记录每个学生上一次收集的 commit 记录的 ID，以此进行筛选。但是这样的方法在实际实验的过程中

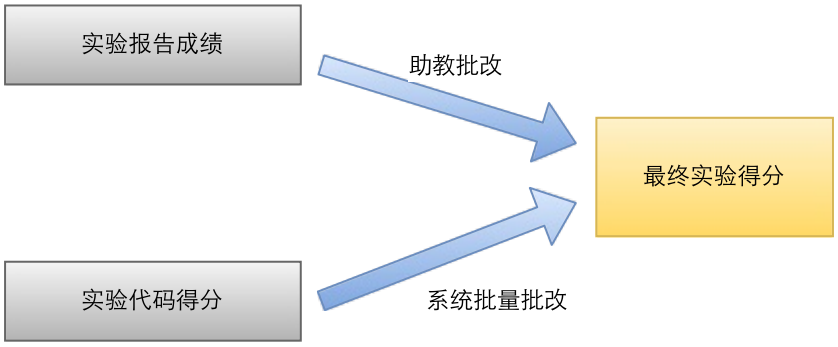

图 3.10　在线实验批改流程

被我们放弃了，因为操作系统课程平台平台上的学生数量较大，如果每个学生都要记录收集状态，带来的开销太大，同时，在开发和调试过程中，一旦出现错误，就要花费大量的时间去人工检查每个学生的数据是否正确，这样的工作量是不太合理的。作为收集者，应该把主要工作放在信息收集上，尽可能简单的完成收集任务。我们最后决定的方法是，每次都获取所有提交，但是不对新提交和旧提交做筛选。将筛选的工作留给消费者处理。这样既能保证效率，又能降低开发难度。

3.3.2　在线实验成绩收集

在介绍在线实验成绩的收集工作前，我们先介绍在线实验的提交和批改方式，实验成绩的主要组成参照图 3.10。学生通过操作系统课程平台上的链接来获取在线实验的代码框架，并根据课程的要求补充里面缺失的代码。在这个过程中，学生将完成的实验上传到 Gitlab 中，在每个实验的截止日期前，学生都可以继续上传代码覆盖原有的版本。实验的截止日期到达以后，系统会根据学生在操作系统课程平台平台上维护的信息，到对应的代码库获取学生的实验代码，并在实验机器的 docker 环境下完成代码的编译，运行和评分工作。系统的评分结果会储存到一个文件中，接下来，助教再依次查看学生的实验报告，并根据实验代码的运行成绩确定最后的分数。无论学生使用 Github 还是 Gitlab 保存自己的代码，系统都可以下载到对应的代码，因此不同的代码管理平台并不影响在线实验的批改。

实验成绩包括代码运行成绩和实验报告成绩。与收集 Commit 信息不同的是，在线实验成绩的收集并不需要使用 Gitlab 的 API，因为最后的结果是由助教维护的，因此实验的成绩会保存到文件里，因此，这一方面的收集者通过读取文

件来生成虚拟的事件数据。

例如，收集者读取最终成绩文件的某一行，了解到学生 A 的实验的得分为 8 分，接着收集者根据这些信息生成一条事件，这则事件数据描述的内容就是系统将学生 A 的实验一最终得分设置为 8 分。我把这些类型的事件成为虚拟的事件，因为这些事件不是来自具体的平台的，而是通过助教运行收集者收集指定文件而产生的事件。这些事件和其他事件一样会发送到事件池中，对于分发者来说并没有明显区别。

3.4 Piazza 平台的数据收集

Piazza 平台是操作系统课程指定的讨论平台，学生在这个平台上的行为表现了其对待学习的积极程度。在 Piazza 平台上，我们会收集学生的提问数和回答数等数据。Piazza 平台也提供了获取这些数据的方法，我们的 Piazza 收集者通过 Piazza 平台提供的 API 来收集平台数据并生成事件数据。

和之前的收集工作不太一样的地方是，我们没有办法方便的获取到每个学生每一次发帖的时间和内容，因为 Piazza 并没有提供类似的 API。但是 Piazza 自己对学生的数据进行统计和整理，同时提供了获取统计数据的 API。通过调动这个 API，我们可以直接获取统计结果，但是这样的方法和我们之间收集事件的设计相互矛盾，因为我们我无法将统计的结果有效的还原回事件本身。也就是说，在之前的设计中，收集者只能将收集到的数据发送到事件池，不能发送到结果池，但是我们收集到的 Piazza 统计数据更适合发送结果池，因为统计过后的结果很难还原为具体的事件。

对于这个问题，我最后采用了以下解决方法。并不将 Piazza 的统计结果还原为事件，而是产生一个新类型的事件，这个事件的描述是：Piazza 收集者从 Piazza 平台上收集了所有学生的行为数据。接下来，在事件的描述中保存收集到的所有信息。对于系统来说，这是一个“大事件”，因为这是对所有学生数据的全部更新。

Piazza 平台的增量更新 由于 Piazza 平台提供的数据已经是处理过后的结果，因此并不需要考虑增量更新的问题，我们只需要让新的结果替换之前的结果即可。为了让这个结果和之前的事件处理模型兼容，我们在对应的事件数据里引入了收集时间的字段，在后期的事件处理中，我们通过收集的时间判断数据新

和旧，然后决定是否用当前数据替换之前的数据。

3.5 总结

在这一章中，我们通过讨论在 Open edX 平台，Gitlab 平台和 Piazza 平台的数据收集，列举了几种类型的数据收集方法。下面列举了我们收集到的主要的数据：

在 Open edX 平台上，我们收集到的数据包括：

- 学生在其他平台上的账户信息
- 学生进行在线练习的时间，答案内容，尝试次数等信息
- 系统对选择题的自动批改时间，自动批改结果
- 在线练习问答题的人工批改结果，评语等

在 Gitlab 平台上，我们收集到的数据包括：

- 学生每一次实验提交的时间，以及对应的 Commit 信息
- 学生的在线实验运行得分，以及对应的错误报告
- 学生实验报告的得分情况

在 Piazza 平台上，我们收集到的数据包括：

- 每个学生的发帖数，提问数，回答数，浏览数以及在线的天数。
- 前五个最佳的提问者，回答者等。

现在我们收集到了足够多的数据，接下来就是对这些数据进行分析和处理。除了进行常见的统计外，我们还可以根据数据的意义对每个学生的得分进行排序和比较，挖掘数据中的更多的含义。在第 4章中，我们将详细讨论分析这些数据的方式。

第 4 章　数据分析

数据处理模块中的主要角色是分发者和消费者，分发者负责对具体的事件类型调用不同的消费者进行处理。消费者则负责将事件处理成具体的结果。

4.1　数据分发

分发工作主要由分发者负责。分发者的工作比较简单，它管理着事件池与消费者。分发者也是一个定时运行的程序，每次运行时，分发者会检查事件池中新增的事件数据，并完成对时间数据的分发。图 4.1展现了分发者的工作流程。

正如表 3.1所列举的，我们的每一个事件数据都指定了一些固定的字段。对于分发者，只需要关心 collector 字段和 topic 字段。每个消费者都需要进行相关的配置，设置订阅事件的来源和主题，这里，事件数据通过 collector 字段来表示其来源，通过 topic 字段表示其主题。每个消费者可以同时订阅多个来源和多个主题的事件，同一个事件也可以同时被多个消费者处理。例如，统计全班成绩分布的消费者和统计某个学生的消费者可能会同时订阅主题为“批改”的事件；统计每个人成绩的消费者，不仅会订阅来自“在线练习”的成绩事件，也会订阅来自“在线实验”的成绩事件。

借助分发者的工作，我们的消费者就可以专心于处理数据而不用关心数据的获取方法。

4.2　数据处理

数据处理过程就是事件的消费过程，我们的学习分析系统通过将一系列的事件数据整合生成我们需要的结果数据。

4.2.1　消费者

数据的处理是由一系列消费者负责的，每个消费者都从分发者接收数据，并进行自己的处理计算。生成对应的数据。一般情况下，一个消费者只负责一个类型的数据处理。例如统计全班的成绩分布和统计每一题的答案分布会分别由不

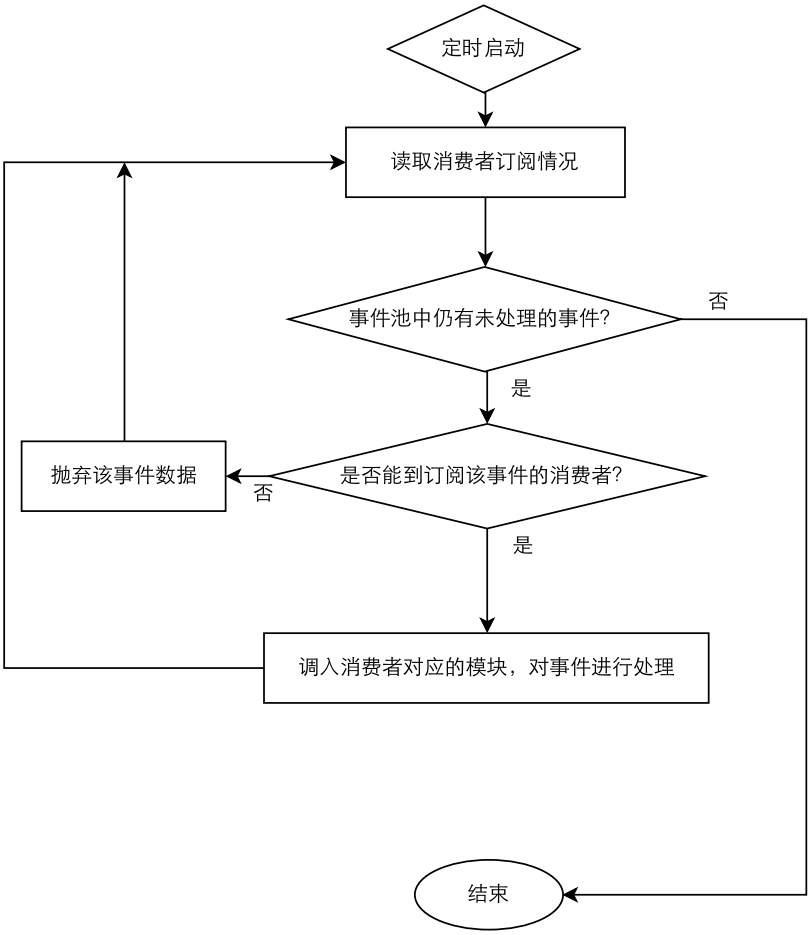

图 4.1　分发者工作流程图

同的消费者处理。这样设计的初衷是为了让单个消费者的只能尽可能简单，以方便后期的修改扩展。举个例子，如果以后我们系统不再需要某种类型的数据，那么我们可以简单的删除对应的消费者即可；如果我们需要增加新的数据处理类型，只需要新建对应的消费者，同时在分发者配置对应的订阅信息即可。

与收集者不同，消费者并不是独立运行的。消费者由分发者在进行数据分发时调用。为了简单起见，在我们的系统的数据处理过程中，一个时间段只有一个消费者在工作。

4.2.2　消费者管理模块

顾名思义，消费者管理模块是一个管理所有消费者的模块。它的目的就是为了满足以下的需求：

第一，虽然消费者在处理每个事件时都会产生一定的结果，但是很多时候

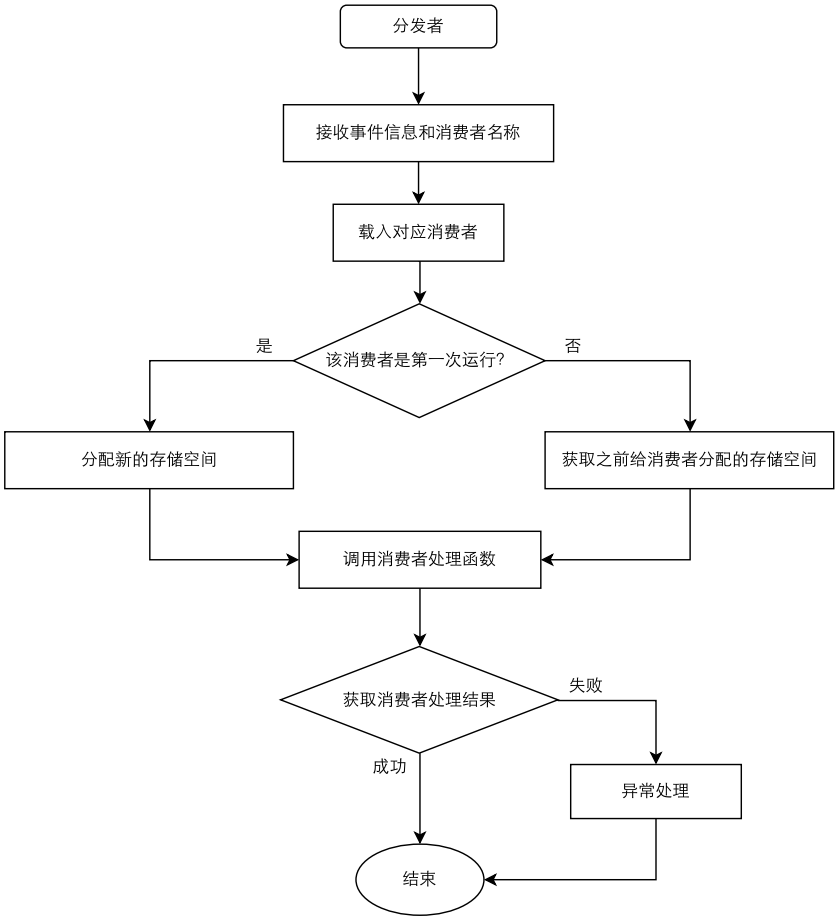

图 4.2　消费者管理模块工作流程图

消费者需要处理大量事件才能得出有意义的结果。例如统计全班同学的成绩分布，消费者需要统计一系列的事件才能得出最终的结果。而实际上这些事件很可能不是连续发生的，因此处理类似数据的消费者需要对自己的状态进行记录，同时对处理的中间结果进行保存。

第二，随着我们系统的发展，消费者的数量会越来越多。如果直接通过分发者根据事件来源和主题来调用消费者进行处理，就要求分发者了解每一个消费者的调用方法，这会导致分发者负担过重，因此我们需要在分发者和消费者之间建立一个统一的通信方式。

我们的消费者管理模块主要负责和分发者和消费者之间的交互和调用。分发者将事件信息和对应的消费者名称发送给消费者管理模块，消费者管理模块根据调用的消费者信息，给消费者在结果池中分配相应的储存空间。消费者则可以借助这个储存空间进行自己的数据更新和维护工作。图 4.2展现了消费者管

理模块的工作流程。

4.2.3 数据处理结果

使用第3章收集到的数据，我们可以处理得到很多的显示结果，例如表 4.1是我们在期中时分析操作系统课程平台平台上的练习批改信息得到的结果。这是一个简单的例子，我们统计所有批改的得分事件，并求出其平均值。从这些数据，老师可以了解到当前时刻班级的整体情况，例如，根据答题人数我们可以发现，大部分同学都跟随着进度完成了约一般的进度，还有少部分人超前完成了所有章节。

表 4.1 班级每章节得分结果

章节	答题人数	平均得分
Section-0	0	NaN/NaN
Section-1	171	10.47/15
Section-2	179	4.88/19
Section-3	146	11.88/19
Section-5	142	3.46/6
Section-6	133	3.40/5
Section-7	126	9.38/11
Section-9	122	5.65/9
Section-10	112	4.77/5
Section-11	114	6.68/7
Section-12	112	3.79/4
Section-15	103	5.92/6
Section-17	95	5.88/6
Section-18	84	5.85/6
Section-19	15	2.80/3
Section-20	14	6.50/7
Section-21	8	7.00/7
Section-22	8	4.13/5
Section-23	8	5.25/7
Other	170	2.51/3

除了全部的整体情况，我们还可以获得学生的数据处理情况，例如表 4.2某个学生在每一讲单独的答题情况结果，为了方便让每个学生了解自己在整个班

级中的状态，我们和全班的数据进行了比较。每个学生只能查看自己的结果，老师可以查看所有学生的结果。

表 4.2　学生每章节得分结果

章节	得分	备注
Section-1	8	185 人答题，有 148 人得分更高
Section-2	6	198 人答题，有 0 人得分更高
Section-3	10	166 人答题，有 125 人得分更高
Section-5	4	154 人答题，有 0 人得分更高
Section-6	3	146 人答题，有 105 人得分更高
Section-7	11	141 人答题，有 0 人得分更高
Section-9	4	141 人答题，有 100 人得分更高
Section-10	5	134 人答题，有 0 人得分更高
Section-11	7	136 人答题，有 0 人得分更高
Section-12	4	134 人答题，有 0 人得分更高
Section-15	6	134 人答题，有 0 人得分更高
Section-17	6	131 人答题，有 0 人得分更高
Section-18	6	129 人答题，有 0 人得分更高
Section-19	2	119 人答题，有 113 人得分更高
Section-20	7	124 人答题，有 0 人得分更高
Section-21	7	126 人答题，有 0 人得分更高
Section-22	4	118 人答题，有 49 人得分更高
Section-23	6	124 人答题，有 0 人得分更高

通过这些数据老师可以详细了解到学生的状态，但是光有数据并不能展现结果的所有信息，例如，我们很难通过表 4.1和表 4.2来比较学生得分和平均得分之间的情况。对于这些数据，我们还需要结合可视化模块进行进一步处理，我们将在第 5章详细介绍。

4.2.4　结果池

结果池相对于我们系统的位置可以参考图 2.1。结果池保存我们学习分析系统的处理结果。同时，消费者管理模块给消费者分配的存储空间也来自结果池（中间结果也是结果的一部分）。我们系统的可视化模块以结果池为输入，这一部分的内容我们将在后面的章节讨论。

消费者的结果都以 json 文件的形式保存在结果池中，用户通过指定路径的方式来获取每一个统计结果。我们数据处理的大多数结果都是为了我们的可视化模块而设计的，我们将在第 5章对处理结果的格式进行介绍。

第 5 章　数据可视化

我们有了数据的统计结果，但是让老师仅仅依靠这些数据内容是很难理解数据表示的全部意义的，例如每个同学在学习过程的学习曲线，某一题各个回答所占的比例，类似的统计需求还有很多。为了展现这些数据反映出的内容，我们需要使用图表对这些数据进行可视化处理。

5.1　可视化模块

可视化模块处于我们系统中的最后一层，老师和学生通过可视化模块来访问我们的数据处理结果。图 5.1是我们可视化系统的的架构图。

关于可视化结果的输出格式也是一个需要讨论问题。我们最后选择将可视化的结果处理为 HTML 的格式。这样的选择主要是出于以下几点的考虑：第一，HTML 是目前非常普遍的数据格式，用户可以直接通过网络，在任意地方浏览我们的结果。第二，搭配 js 脚本，我们的图标可以提供非常强大且友好的交互功能。第三，在网络上有很多基于 HTML/CSS/JS 的第三方可视化库，我们可以借助这些可视化库来减少我们的工作量。

我们的可视化模块以结果池的数据作为输入。经过一系列处理，将结果转换为 HTML 格式的数据展现给用户。可视化模块需要基于结果池来工作，因此从图 5.1我们看到结果池处于最底层，但是实际上结果池并不属于可视化模块。只要文件的格式满足要求，任意的数据库都可以成为结果池，不过为了方便起

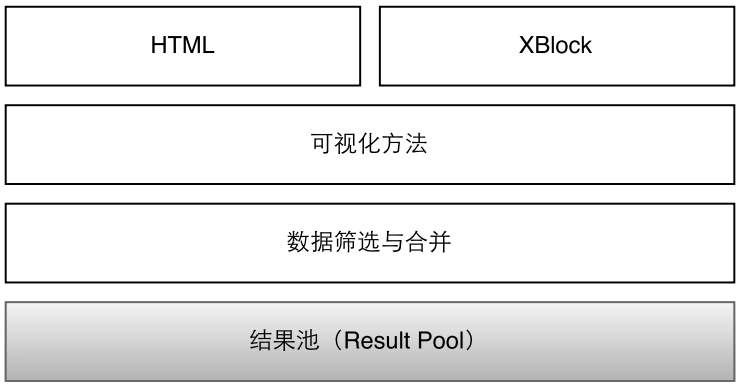

图 5.1　可视化模块架构图

见，我们将结果池归到可视化模块的第零层。

可视化模块的第一层负责对结果池的数据进行筛选与合并。考虑这样的需求：有时候我们需要通过比较不同来源的数据来得出一些结论，或者，有时候我们需要对数据进行一些简化，剔除我们不感兴趣的数据，只展现我们感兴趣的内容。例如，有时候老师需要比较学生 A 和学生 B 在各个方便的得分情况，或者比较一个学生在全班中的位置。像这种时候，我们就需要对结果池的数据进行第二次处理，也就是筛选与合并。这一层根据用户的需求，从结果池中获取对应的数据，并对数据进行简要的筛选，如果用户选择了多组数据，则还需要对获取的数据进行合并处理。

可视化模块的第二层是负责将数据转化为对应的 html 代码。目前有很多基于 js 的前端可视化库，我们选择了 HighCharts.js[6] 作为我们的可视化库。HighCharts 是一个各方面较为优秀的可视化库，提供多种图表的支持，同时 HighCharts 有不错的外观以及良好的交互方式，并且提供较方便的开发方式以及强大的社区支持。

可视化模块通过结果池获取消费者的处理结果，但是这个过程中也有一些需要注意的问题。每个消费者产生的处理结果的类型很可能是不相同的，可视化模块很难保证对每个类型的数据进行恰当的处理。因此我们需要对结果池的数据进行一定的限制。在结果 json 数据中指定数据的类型，以及希望使用的渲染方式。这样我们就可以通过一些相关的字段来确定需要使用的可视化方法。图 5.1是结果池的数据说明，图 5.2是我们结果数据的一个实例。

表 5.1　结果池 Json 数据的主要字段说明

字段名	类型	描述
visualization	string	优先使用的渲染方式
stat	json object	消费者处理的结果
title	string	标题
xTitle	string	(可选) 图表的横坐标名称
yTitle	string	(可选) 图表的纵坐标名称
lastMdf	utc string	最后的更新时间
type	string	数据的格式类型

我们统计了学习分析系统的数据收集类型，发现我们可以将大部分图表都

```
{
    "visualization":"pie",
    "stat":[
        {"y":152,"name":"B"},
        {"y":4,"name":"C"},
        {"y":14,"name":"A"},
        {"y":1,"name":"D"}
    ],
    "title":"第1132题答案分布图",
    "lastMdf":"2016-05-17:17:48:12",
    "question":{
        "type":"single_answer",
        "q_number":1132
    },
    "type":"CountStat"
}
```


图 5.2　结果池数据示例

归类为同一种格式，例如直方图，折线图，和饼图。我们都可以使用一系列的二维坐标来储存这些图标需要的数据。例如图 5.2展示的是对题号为 1132 的题目的答案选择分布。我们将每个数据点都转换成结构相同的 json 对象，像这种类型的数据格式，我们可以同时将其应用到饼图，直方图，折线图等。我们将类似这样结果的数据的‘type’字段统一设置为‘CountStat’，同时我们还有‘visualization’字段，用于指定这则数据优先使用的可视化方法。示例中的数据是选择题的分布统计数据，我们更关心选择每个选项的人数所占的比例，因此在这个例子中我们将其‘visualization’字段设置为‘pie’，代表优先使用饼图进行渲染。

需要说明的一点是，每个结果数据指定的优先使用的可视化方法是可以被用户修改的，老师和学生可以根据自己的需要指定任意合理的可视化方案。

除了上面说到的‘CountStat’类型的数据，我们还有其他类型的结果数据。例如统计回答频率的热力图。这些类型的数据无法使用经典的统计图来表现，我们需要将其设置为其他类型。

第三层是也就是我们可视化模块的最后一层，这一层通过将上一层产生的图表组织为可以通过浏览器阅读的页面。我们的展示方式主要依赖用户的浏览器。不过我们仍然需要将我们的结果展示整合到操作系统课程平台平台中以方便老师和学生进行访问。

因为我们最终结果以 HTML 的形式展现，因此理论上所有支持 HTML 显示

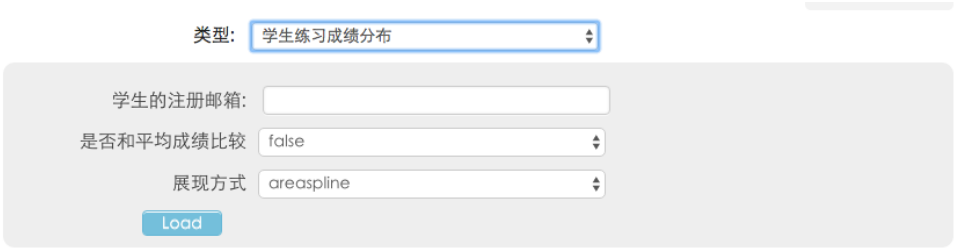

图 5.3　操作系统课程平台上的图表浏览 XBlock

的平台都可以访问我们系统的处理结果，操作系统课程平台也不例外。我们为操作系统课程平台开发了对应的 XBlock。使得用户可以通过简单的选择查看我们的图表，例如图 5.3。用户可以选择不同的数据结果，根据不同类型的数据，填入对应的参数。同样，用户可以选择自己需要的可视化方式来展现图表。这个扩展还有一个作用，就是对用户的请求做权限判断，为了保护学生的隐私，一般情况下我们限定每个学生只能查看自己相关的数据。因此，我们通过这个扩展判定学生的请求，如果学生没有对应权限，则禁止其查看对应数据。

5.2　可视化结果示例

我们的学习分析系统经过收集者，分发者和消费者的处理，从原始数据中提取了很多有用的数据，借助我们的可视化模块，老师和助教可以通过图表直观的清晰的了解到很多事情。例如图 5.4，这是对于题号为 1161 的题目的学生回答统计图。通过对图表的观察，我们可以发现，大约四分之三的学生选择了正确答案 ACD，不过，值得注意的是也有较多的人选择了 AD 作为答案。通过这个图表，老师就可以注意到这一题的 C 选项是一个可能是一个易错点，因此在后期的习题课中就可以把经历放在这一题的 C 选项上。借助我们选择的可视化库，我们为图表添加了一些简单的交互功能，例如当鼠标停止在每一个区块时，会产生对应的提示，显示当前选择这个选项的人数。

我们的系统支持用户根据自己的需求，对同一个数据选择不同的展现方式。见图 5.5。我们使用相同的数据，但是使用柱形图作为展示方法。通过柱形图，我们可以直观的看到选择每个选项的学生数。

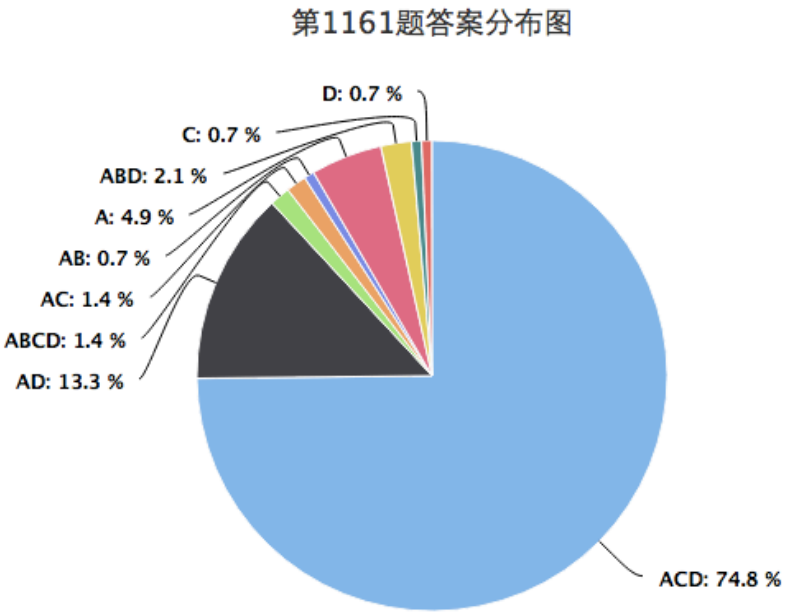

图 5.4　学生回答统计图-饼图

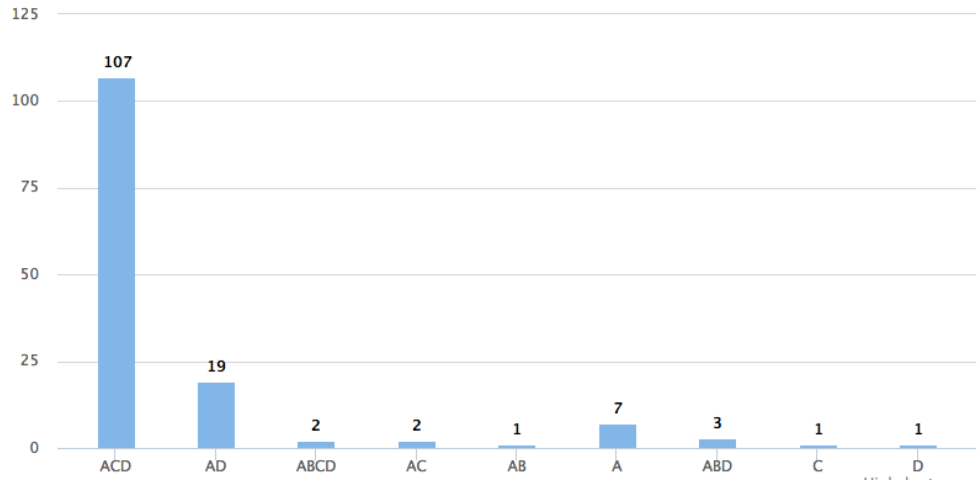

图 5.5　学生回答统计图-柱形图

除了提供常见的可视化方法，我们也提供了其他类型的数据以及对应的可视化方法。例如学生回答情况热力图 (图 5.6)，图表的横坐标轴和纵坐标轴都是时间轴，横坐标轴代表日期，纵坐标代表时间。图表中间是一系列的方块，每个方块代表一小时，方块的颜色代表在这一小时内参与在线联系的人次。白色的方块代表当前时间段无人答题。每个方块的颜色会随着答题人次的增多而不断变化，即从淡黄色过渡到红色。从这个图里面我们可以发现很多事情。例如学生

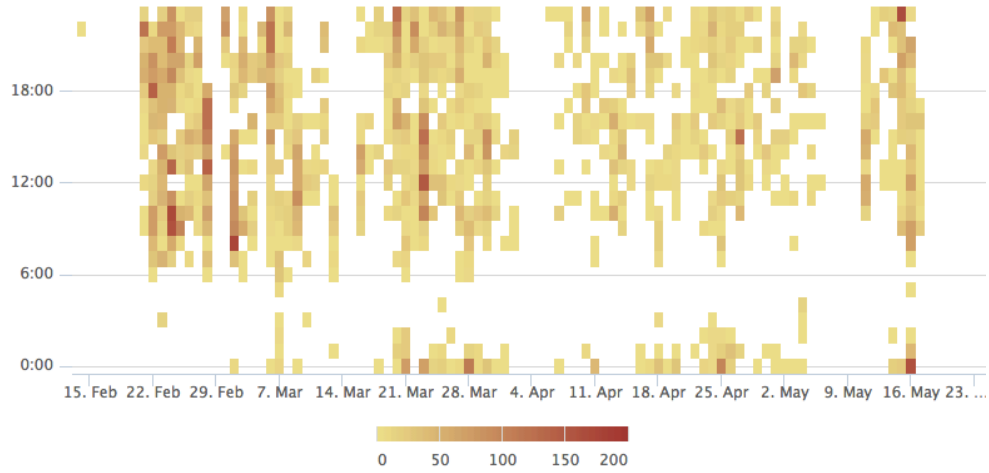

图 5.6　学生回答频率热力图

每一天参加在线练习的时间一般会从早上 6 点开始，在晚上 12 点结束，偶尔答题时间会延长到 12 点之后。

正如之前介绍的一样，我们的可视化模块不仅支持对同一个数据选择不同的可视化方法，也支持对将多个数据同时进行比较。例如图 5.7是单个学生在不同章节的得分分布。图中蓝色的线是学生个人的成绩，黑线是整个班级的平均成绩。因为每一讲的总分不一样，为了让可视化的显示有意义，我们对成绩的显示进行了归一化，将每一讲的得分都映射为 [0, 1] 之间的数值，代表总分的获得程度，例如 0.5 则表示学生获得了 50% 的分数，1.0 则表示学生获得这一讲的所有分数。通过比较个人的成绩和班级平均成绩，老师可以很容易获取每个学生的状态，以及时了解每个学生的状态，而学生个人也可以及时了解自己的状态，以对自己的学习投入进行适当的调整。

5.3　学生和班级的综合报告

除了提供单个图表的可视化，我们还提供了表格的可视化，支持将消费者处理的结果进行表格的数据展现。结合表格和图表，我们可以为每个学生生成对应的报告，这也是我们学习分析系统的最终目标。图 5.8，图 5.9和图 5.10是班级报告的一个示例，我们将从操作系统课程平台，Piazza 平台以及 gitlab 平台收集到的数据整合成一份报告，老师可以直接通过报告了解学生在每个平台的状态，这样老师可以对学生进行一个全面的评判。

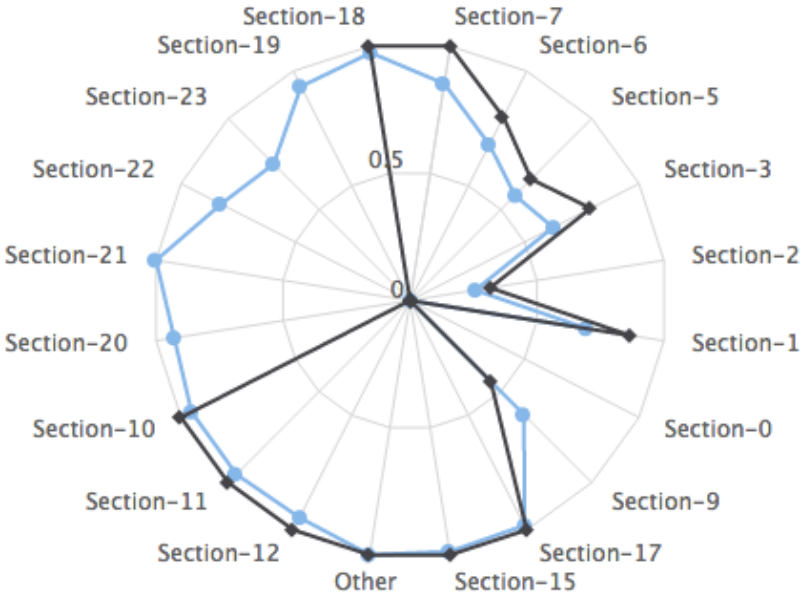

图 5.7　学生各章节得分分布

操作系统课程综合报告

课程信息

清华大学2016春-操作系统

在线练习情况

各章节得分分布（归一统计）

章节	答题人数	平均得分
Section-0	0	NaN/NaN
Section-1	185	10.36/15
Section-2	198	4.84/19
Section-3	166	11.60/19
Section-5	154	3.52/6
Section-6	146	3.44/5
Section-7	141	9.48/11
Section-9	141	5.61/9
Section-10	134	4.75/5
Section-11	136	6.74/7
Section-12	134	3.81/4
Section-15	134	5.90/6
Section-17	131	5.84/6
Section-18	129	5.85/6
Section-19	119	2.92/3
Section-20	124	6.80/7
Section-21	126	6.83/7
Section-22	118	4.33/5
Section-23	124	5.74/7
Other	200	2.78/3

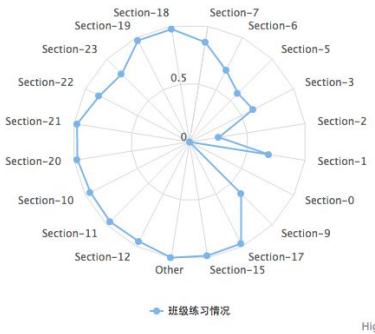

图 5.8　班级综合报告-1

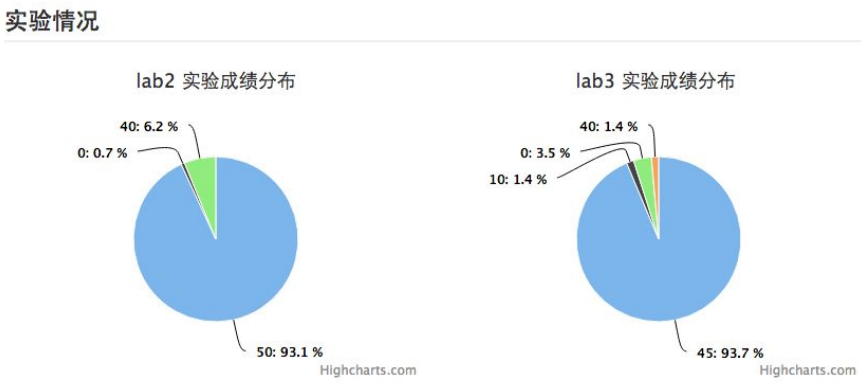

图 5.9　班级综合报告-2

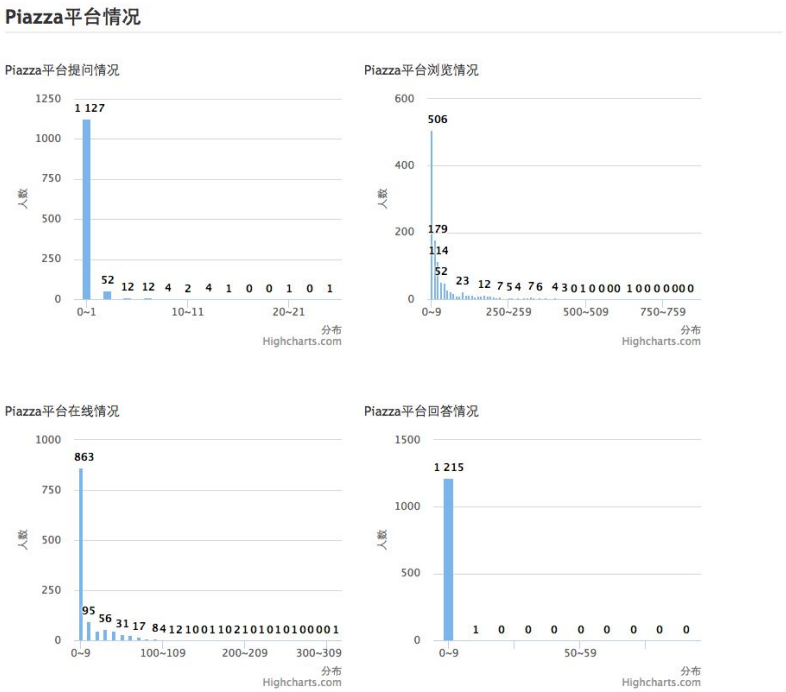

图 5.10　班级综合报告-3

第 6 章　总结

原生的 Open edX 平台虽然自己支持一部分的统计功能，同时也有一些统计工具可以收集更多 Open edX 的数据，但是其不能支持多平台的统计，这让依赖 Piazza 平台和 gitlab 平台的操作系统课程平台的操作系统课程的学习信息收集工作变得十分困难。我们的学习分析系统是一个与 Open edX 独立的数据收集系统，可以同时收集到多个平台的数据，并对其进行整合。让老师可以方便的了解学生在各个平台上的表现情况，根据每个学生的情况和班级整体的情况及时了解课程可能出现的潜在的问题。学生也可以及时了解自己在操作系统课程的情况，以及时了解自己的情况，防止自己因为之前的不注意而出现挂科等现象。

不过我们的系统仍然有很多可以完善的地方，我们现在的工作主要是搭建了一个较为完整的系统，在各个平台上还有许多我们没有收集的信息，同时也有很多信息我们没有提取出来。如果收集的数据足够多，处理足够精细，也许我们还可以根据学生的数据来预测其最终的成绩，对当前表现不太理想的同学进行及时干预。如果能继续完善现有的系统，我们的系统就能够更好的为老师和同学服务了。

插图索引

图 1.1 Open edX 上的学生个人作业完成情况 2
图 2.1 学习分析系统主要架构 6
图 2.2 数据收集和分发 7
图 2.3 数据分析和可视化 8
图 3.1 事件的分发过程 10
图 3.2 事件 json 数据样例 11
图 3.3 原始的 markdown 格式的题目样例 12
图 3.4 升级后的 json 格式的题目样例 13
图 3.5 操作系统课程平台上的题库编辑 xblock 13
图 3.6 升级后的问答系统 XBlock 14
图 3.7 回答数据格式样例 15
图 3.8 操作系统课程平台收集者工作流程图 16
图 3.9 gitlab commit 收集者工作流程图 17
图 3.10 在线实验批改流程 18
图 4.1 分发者工作流程图 22
图 4.2 消费者管理模块工作流程图 23
图 5.1 可视化模块架构图 27
图 5.2 结果池数据示例 29
图 5.3 操作系统课程平台上的图表浏览 XBlock 30
图 5.4 学生回答统计图-饼图 31

图 5.5　学生回答统计图-柱形图 31
图 5.6　学生回答频率热力图 32
图 5.7　学生各章节得分分布 33
图 5.8　班级综合报告-1 33
图 5.9　班级综合报告-2 34
图 5.10　班级综合报告-3 34

表格索引

表 1.1　可视化库的调研结果 ……4

表 3.1　事件 Json 数据的主要字段说明…… 10

表 4.1　班级每章节得分结果 …… 24

表 4.2　学生每章节得分结果 …… 25

表 5.1　结果池 Json 数据的主要字段说明 …… 28

参考文献

[1] https://open.edx.org

[2] Hernández-García A, Conde M A. Dealing with complexity: Educational data and tools for learning analytics. Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality, New York, NY, USA: ACM, 2014. 263–268

[3] Ruiz J S, Díaz H J P, Ruipérez-Valiente J A, et al. Towards the development of a learning analytics extension in open edx. Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality, New York, NY, USA: ACM, 2014. 299–306

[4] Cacatian C T M, Francisco M R C, Jamandra A J T, et al. Use of analytics to improve student behavior and performance in an online course implementation. IISA. IEEE, 2015. 1–6

[5] http://docs.gitlab.com/ce/api/

[6] http://www.highcharts.com/

致　谢

衷心感谢导师向勇副教授对本人的精心指导。他们的言传身教将使我终生受益。

感谢清华大学计算机协同工作实验室全体老师和成员的热情帮助和支持。

感谢在我论文设计时期使用操作系统课程平台的所有同学，他们的在学习过程中产生的数据以及使用平台过程中的意见和反馈为我的工作提供了很大的帮助。

感谢 ThuThesis，这个模板使我后期的论文撰写工作变得轻松许多。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

附录 A　外文资料书面翻译

基于 **Open edX** 的学习分析开发

摘要**:** MOOC（大规模开放在线课程）平台的出现显著增加了大规模学习分析的需求，因为教师很难同时关注成百上千的学生。然而在当下，MOOC 平台对于学习分析的支持仍处于非常初级阶段。edX 虽然是目前最重要的 MOOC 平台之一，但是也几乎没有学习分析的功能。在本文中，我们分析了 edX 现有的学习分析功能以及实现学习分析的主要方法，同时给出在 edX 中实现学习分析扩展组件的初始步骤。然后我们会回顾其中的技术、困难和解决方案、涉及到的架构以及其它相关的事情。最后，我们提供一些新的可视化方法，来帮助教员和学生了解课程在平台上的进展情况。

A.1　引言

使用虚拟教学环境（VLEs）是目前非常常见的教学方法。通过使用现有的技术，我们可以改善传统教育中出现的的问题，同时在各个方面对学习的过程进行优化。例如，教师能根据学生的学习情况来调整教学方法，从而优化每个学习周期的结果，而学生通过检查自己在虚拟教学环境上的表现情况来了解自己在学习中的误区和错误。学生在虚拟教学环境的活动可以产生大量的原始数据，但是，真正读懂这些原始数据是非常困难的，因此，我们需要从这些原始数据中提取用户需要的信息。

为了满足这一需求，产生了很多新的研究领域，学习分析就是其中之一。学习分析的主要内容，就是通过收集各种各样的数据（例如成绩，个人信息以及虚拟学习环境的相关结果）[1]，借助高级建模技术来帮助学生和老师了解教学过程。

MOOC 就是虚拟教学环境的一个实例。借助于大量学生的积极参与，以及丰富的教学资源 [2], MOOC 这几年发展势头迅猛。MOOC 在几年前才开始出现，不过也有人认为 MOOC 的方法其实在很久以前就已经被我们使用，O'Donnel[3] 就是其中之一。许多有影响力的大学和公司，例如哈佛大学和麻省理工学院，

还有谷歌和微软，都为 MOOC 的发展提供了重要的支持。目前网络上有很多的 MOOC 平台，但是在 2013 年，最大的三个平台分别是 Coursera，Edx 和 Udacity[4]。这些平台都有上万的注册用户，同时有数千学生在平台上学习。这样的用户量大大增加了 MOOC 平台对学习分析组件的需求，因为老师无法同时兼顾这么多学生，同时，学生自己也需要借助学习分析组件来了解自己的情况。一个可能的例子是，某些情况下会有很多的学生苦苦挣扎于某一个考试，而这很可能是某个教学上的疏忽造成的。如果学习分析组件可以帮助教员及时发现这些事情，那么学习分析组件对于这个课程就有很大意义。

目前，edX 上有 53 所学校总共 250 万的学生同时在线学习超过 200 个课程 [5]。同时，edX 已经开源，希望创造一个人人都可以参与开发和改进的环境。这个开源软件已经被包装成 Open edX，其源代码可以在 GitHub 上找到。然而，edX 平台上对于学习分析的支持仍处于最初级的阶段，我们仍然有很多的工作需要完成。目前 edX 已经放出了第一次数据分析报告 [6]，但是教员和学生仍然需要一个更强大的学习分析组件，以改进 MOOC 课程的体验。

要在 edX 上实现学习分析仍然需要解决许多问题，一个问题是，如何在考虑数据存储，APIs 和基础结构的同时，将 edX 和学习分析技术结合起来；另一个问题是，在 edX 特殊的环境下，应该选择什么样的指标、计算方式以及可视化方案。

在本文中，我们提出了开发 edX 学习分析扩展组件的第一个步骤，其中包括了对不同技术方案和问题的分析、如何与通过 EDX 架构整合以及研发过程中涉及到事情，我们还提出了一些新的可视化方案。

文章将按照以下顺序介绍我们的工作。首先，我们将在第 2 节介绍关于学习分析和可视化的一些相关工作，第 3 节我们分析 edX 本身对的学习分析的支持，第 4 节我们介绍学习分析扩展组件的技术和结构，而其中涉及到的可视化技术我们会在第 5 节单独介绍。我们在第 6 节总结现有工作和介绍未来的工作。

A.2 相关工作

选择有意义的指标是实现学习分析组件最大的挑战之一。这些指标一般是原始数据通过计算产生的结果。而指标的定义和计算取决于具体的平台和教育的环境。Dyckhoff 曾经对不同的学习指标进行调查和比较 [7]。针对特定平台来

确定指标的例子在 [8] 给出，Khan 学院平台给出了一些更高级别的信息指标（例如效率，不经思考就查看提示的学生，粗心大意的学生）的定义。为 edX 开发学习分析组件，还有几个难点，一个就是确定 edX 上有意义的指标，还有一个难点就是确定最优的计算方法。

利用这些指标和原始数据，我们可以使用各种各样的可视化方案让教员、学生可以很容易地了课程的进展情况。在这一方面，Student Activity Monitor（SAM）[9] 支持教师和学生信息表格的可视化，而 CAMera[10] 则可以对语境相关的元数据进行可视化。其他的平台，如 Moodle[11] 也有一些学习分析的支持，它们提供过滤功能，用户可以根据操作类型、时间和具体活动信息对原始数据进行过滤。虽然 Moodle 没有对数据可视化的功能，但是 [12] 里面提到的工作也可以让 Moodle 拥有不错的可视化效果。

Khan 学院平台 [13] 也支持不同的信息可视化，例如每个学生的技能学习情况、学生获得的成就，花在视频和练习的时间，还包括一些普通交互的可视化。然而，它的可视化功能不支持其他的指标。出于这个原因，一个 Khan 学院平台的学习分析组件 ALAS-KA 产生了。ALAS-KA 提供其它信息的可视化 [14]，但是它并不是 Khan 学院的原有功能。

在 edX 上实现学习分析结果的可视化，还有一个重要工作就是分析其它平台目前正在使用的可视化方案（例如 Khan 学院平台），以便选择相似的方法。当然我们也需要根据数据的特征考虑使用新的可视化方案，让数据可以更好的被使用。

此外，在 edX 上开发学习分析组件时也有许多技术细节要考虑，例如存储原始数据的方法以及用于访问数据和 edX 结果的 API。同样，一些需要在其它平台需要处理的技术问题（例如 Khan 学院平台），在 edX 平台上也可能会不一样。此外，edX 中的一些技术细节也在其他场合被讨论过，例如关于严肃游戏的讨论 [16]。

A.3 edX 上的学习分析

本节将讨论 edX 学习分析中涉及到的信息。首先，我们讨论 edX 在学生交互过程中产生的原始数据。然后，我们总结一下 edX 现有的学习分析功能以及一些改进的方法。

原始数据

edX 上的课程通过安排教学活动和教学资源来进行教学。在 edX 平台的数据核心可以找到很多信息，课程和学生则是这些信息的两大来源。课程信息被分成结构信息和内容信息。你可以把课程结构当作一棵树上的一个分支，上面有很多叶片，而叶片代表内容。在自上而下的层次结构中课程被分为章，章的下一级是节，节则是课程内容依顺序排列的地方。上面提到的每个节点（这包括所有的树枝和树叶）在 edX 中被称为模块。edX 也可以把课件单元的集合当做一个模块。每个模块的状态就是一个键值对的集合。课程信息在课程开始之前就已经创建，学生信息则在课程进行的时候产生。当一个学生使用课件的一个模块，如果这是第一次访问，一条状态数据将被创建，否则这条数据将被更新；这使得 edX 可以记录每个模块与学生的状态，这对于学习分析组件有很大意义。状态数据的字段和类型示于图 1 的 courseware_studentmodule 表中。我们可以用视频举一个例子，状态数据可以保存学生上次观看视频的位置和播放速度。

此外，还有另一个数据流，我们可以把它称作跟踪数据。几乎学生和教员每一次和 edX 交互都会产生一条跟踪数据。对于学生来说，跟踪数据主要来源于注册，登入登出，视频和教材的读取，在线练习以及论坛活动。储存跟踪数据的表格示于图 1 的 track_trackinglog 中。在图 2 中可以给出一个跟踪数据具体的例子。通过处理这些原始数据我们可以获得有用的信息，例如，如果有检测到大部分学生在视频的某个位置都选择暂停或重播，那么这个地方就很可能会成为一个考试的要点。这只是一个例子，但是很容易举一反三。

对于学习分析来说，学生与 edX 平台交互产生的大量信息其实是一个潜在的金矿，但是这些数据本身并没有什么特别之处。要使这些数据对教员和学生有用，我们需要强大的预处理能力。学习分析组件可能会成为一个非常强力的工具，它使最佳决策变得可能，同时还可以显著改善教学体验。

edX 上对学习分析的支持

目前 edX 支持基本的学习分析。edx 上有一个页面可以为学生展现自己的课程进展情况，学生能够在学习过程中查看自己进度的变化。页面由两部分组成：图表以及文本信息。这里说的进展情况可以等价于学生课程的完成情况。edX 上的教学进度考虑了作业，实验等情况。在这方面上，edx 没有评分的方法。图 3 是这个进展情况图表的一个例子，通过一个条形图来展现每章问题的完成情况

的百分比。在水平轴上的 Ex01，Ex02 和 Ex03 指章节的顺序数，这里只统计了当前参与课程的学生情况。edX 正在开发的一个测试版的统计表可以向教员显示更多的学生分布信息，例如每题分数，出生日期，性别和教育水平。大部分的这些信息都可能成为人口学和社会学的兴趣点，但是为了保持篇幅，我们还是继续关注学习分析。

在 edX 上实现学习分析的方法

edX 团队意识到将原始数据转换成关于学生和教员 5 的信息的需求。edX Analytics Service6 项目和 Insights5 项目就是 edX 团队在这上面的努力。Insights 是一个 Python + Mongo + Django 的框架，根据 GitHub 上库的描述，这个框架支持基于流事件的简单，可扩展的分析。但是这两个项目目前都尚未完成。Insight 是一个独立于 edX 的系统，设计为通过读取数据库来获取 edX 的信息，它提供了一组 API 去实时的处理那些被触发的事件。近日，edX 团队一直在致力于 edX 分析 API 客户端 7 的开发，这是一个工具，允许用户获取 EDX 数据仓库中的数据。

还有一个 edX 的实施 A / B 测试的方法 [17]。这些方法是给人两个过程颇为相似课程，但改变其中的一些功能（例如资源的顺序），并比较哪个版本能更好地工作。以此方式，也可以改进课程。A / B 测试计划也提出了一些可视化方案被提出，这些方案用于比较两个不同的群体的学生的信息，例如展现学生在课程中的路径。此外，MOOC 平台的实时分析已经有了一些可视化的例子 [18]。对于 edX 中视频学习分析的可视化方案的具体实例也已经提出 [19]。尽管大家不断努力地去为 edX 提供各种各样的学习分析支持，edX 的学习分析功能仍然有很大的提升空间。我们觉得，利用大量原始数据来给 edX 的用户提供信息的学习分析的组件是很有意义的。

A.4 技术和结构概述

子小节 4.1 是 openedx 架构的回顾，而第 4.2 节介绍了我们对学习分析扩展的提议。

edX 架构

edx 主要有两部分组成：内容管理系统（CMS）和学习管理系统（LMS）。CMS 是一个用于创建 edX 课程的编辑工具，LMS 则是进行教学的环境。这两个系统与 edX 其余部分的连接几乎是一样的。我们的学习分析组件会着重于 LMS，因为它和我们的目的更相关。

LMS 的主要原则是模块化。不同的功能由各个模块实现。这种模块化的结构使开发新功能变得更容易。每个模块通过组装，可以与其它模块或系统交互。我们的学习析组件适合这个环境，我们会在下面讨论细节。

课程数据被划分为课程内容和课程结构，以方便重复使用。课件由 XModules 和 XBlocks 模块化，XModules 是平台原始的概念，但为了克服一些固有的限制，它们被由 XBlocks 取代。XModules 和 XBlock 可以存储状态，渲染并展示自己的内容，同时处理用户操作。XBlocks 是比 XModules 更强大的工具，支持第三方扩展，专门为开发者准备。开发人员只需要实现新的 XBlock 实体（他们设计容器），然后教员负责创建内容或重用现有的实体（他们填充容器），学生在学习过程中（它们使用容器）使用这些实体。

技术

edX 基于各种技术的支持，但是这些技术大多没有足够的文档说明。这个问题对于新的 openedx 开发者来说是一个门槛。不过随着时间推移，通过从论坛获得开发者的反馈，edX 团队不断完善现有系统，现在开发环境的安装和使用已经得到显著改善。但是安装开发环境的过程仍然不是全自动的，你可能需要参考论坛上的解决方法，同时需要依靠自己的专业知识和经验。Virtual Box 是用于管理开发环境的工具，Vagrant 则是是构建开发环境的工具。它们为开发者提供一个便捷容易的方法来复制开发环境，以便能够在要求的条件下进行开发。

edX 基于 Django 框架的一个项目，Django 项目都由各种功能单元组合而成，这些单元即 Django 应用程序。Django 是由 Python 实现，要参与 edx 开发，你需要了解这个编程语言。同时，考虑到 edX 是一个 Web 应用，它也包含其他网络技术如 HTML，CSS 和 JavaScript，还有 Mako 模板。

数据持久化

edX 使用两个数据库引擎：MySQL 和 MongoDB。MySQL 是由表组成的关系型数据库，MongoDB 保存 BSON 文档（二进制的 JSON 对象）。与课程相关的信息保存在 MongoDB 中。学生相关的数据存储在 MySQL 的表格中。

Django 的原生支持 MySQL。它默认配备一些内置的表，储存用户认证的信息。Django 提供了 MySQL 数据库操作的 API，但是 Django 的本身并不支持 MongoDB。因此，为了访问存储在 MongoDB 中的数据，开发人员可以使用集成相关功能的 XModule 或 XBlocks 的 API。对 MongoDB 的访问，也可以通过 Python 库 pymongo。一些免费的可视化工具也可以用来从 edX 外界访问数据库，如 MySQL Workbench 和 Robomongo。

我们的学习分析组件

我们可以在图 4 中看到我们的学习分析扩展在 edX 架构中的位置。它是一个新的模块，但是可以和之前已经存在的模块很好地兼容。正如我们之前讨论过的的，edX 架构非常复杂的，因此这个架构图是一个简化的版本，省略掉了与我们学习分析不相关的部分。

edX 应用借助 Django 框架来实现其中大部分内容，这使得 edX 成为一个非常模块化的应用，可以重复使用。多亏这种模块化的设计，我们在开发学习分析组件的时候能够重用 edX 的许多模块。我们的学习分析组件是 LMS 中一个 Django 应用。借助 edX 已经包含的读取数据库的 API，我们在第一层实现了一个新的 API 来获取原始数据，这个 API 能够从数据库查询数据，不需要关心数据储存在 MySQL 上还是 MongoDB 上。借助这个 API 我们可以很方便地获取课程数据和学生数据，包括学生的个人信息和分数，以及学生的跟踪数据。换句话说，第一层的主要目的是提供一种简单的方式来组织和检索数据。

学习分析组件的第二层以上一层获取到的数据作为输入。这一层负责将原始数据处理成有用的信息。对于追踪数据，我们需要充分处理，以发挥其潜在优势。例如，可以通过收集的事件发生的时间点信息来分析学生在每个章节中的信息，通过分析可以得到很多信息，例如学生是否是根据教员安排的顺序学习各个章节。这样我们可以得到每个学生在每个活动或者每个章节中所花费的时间。这是我们学习分析组件最重要的一层，因为它将无意义的数据转换成可以利用的信息。文章中的许多功能都在这一层实现。

最外面的一层负责可视化的处理和呈现。这一层接收之前处理的数据。用户可以在他们的顶部菜单中找到一个新的标签，通过这些标签可以访问这些可视化效果。我们大部分的可视化是通过 JavaScript 实现，使用 Google Charts API 8，支持交互的访问方式。Google Charts API 提供了很多易用的图表实现。除了支持与用户的交互，还可以提供额外的信息，这可以改善用户体验。

安装和使用

学习分析的扩展很容易在任何 Open edX 平台使用。开发者只需要在 LMS 系统中引用 Django 应用，模板和静态内容文件夹在，并将应用程序添加到他们 Open edX 的配置文件中。

使用这个扩展也很简单，用户在他们的课程页面会看到一个新的标签，让他们可以根据角色（学生或教员）来设置不同的可视化方案。学生也可以访问使用自己的数据生成的可视化结果，一些可视化结果也会同时全班的结果，让学生可以了解自己在班级中的位置。

当然，教员能够访问更多的可视化结果。一方面，他们能够访问每个学生的可视化结果，另一方面，它们可以查看更多的统计结果，这对于课程反馈是十分有帮助的。考虑到 MOOC 平台上的学生数量可能非常大，有时候一部分学生的统计结果对于教员了解学生情况来说并不十分有意义。为了解决这个问题，我们的可视化系统将为教师提供过滤和排序功能，让教师可以根据分数，年龄等属性来对数据进行过滤。

经验和教训

为了捕获用户使用平台时的事件数据，edX 团队投入了大量努力。很多事件都会被加入追踪数据中，借助 API，我们可以很轻松的获取这些数据。因此，edX 平台有足够的条件开发更强大的学习分析组件，但是有很多前序工作还没有完成。不过，这些促进学习分析的工作都将在未来几年内完成。

虽然 edX 模块化提供的开发方式很容易使用，但是在具体的开发过程中，我们仍然遇到了一些问题。第一个问题也是大多数开发者都会碰到的问题，就是 edX 项目的容量。edX 是一个很大的项目，其中大部分模块都是有文档的，但是一些较新的模块却很难找到文档。这个问题的原因比较复杂，总的来说是因为平台的发展，导致模块的频繁变动和 API 的频繁变动。这在开源软件中十分常

见，但相比之下，开源软件有许多优点。为了解决这些问题，我们建议开发者在开发开始之前研究平台的代码，以了解平台的代码结构。了解平台的架构将帮助你设计你自己的扩展模块，同时可以很容易地让你的代码适应 edX 平台未来的任何变化。

我们的一些可视化的需要处理大量数据，这是我们不得不面对的另一个问题。因为 edX 平台是一个 MOOC 的平台，所以我们将有大量的数据处理的工作，这就决定了实时统计并展现所有学生的数据是不可能的。我们发现解决这个问题最简单的办法是使用一个任务调度工具，在合理的间隔时间内来统计信息。在我们的例子中，我们决定使用 Celery，因为它容易与 Django 的集成，同时，在 edX 项目的其他部分也使用了 Celery。希望为 edX 贡献自己代码的开发者都会遇到一些问题，但是采用这种方案，潜在的问题可能被避免。

A.5 可视化实例

MOOC 平台产生大量的原始数据，这些数据无论是教师还是学生都很难理解。我们正在开发的学习分析组件的主要目的就是是给教师和学生一个简单的方法来处理大量的数据，同时用简单和方便的方式呈现处理的结果。个人信息的收集也将包含在我们学习分析之中，作为一个对教师和学生有意义的反馈。我们将根据课程相关的信息例如练习，视频，教材以及学生的交互，来提供不同类型的可视化方式。

通过使用学习分析组件，平台的用户可以在很多方面上受益。教师可以通过一个图表来了解整个课程的进展情况，或者通过统计信息来发现一个影响整个课程的问题。此外，他们还可以单独监控每个学生个体（或每个群体），如果发现有些学生表现不佳，必要时可以进行干预，从而防止学生中途退出或者挂科。学生也可以很好的利用这个扩展，主要是通过提高他们学习时的自我意识，同时比较自己与课程其他学生的结果的差异。

不同学生任务的进展情况分布

第一个可视化是显示学生在同一门课程每个章节的不同得分的分布的条形图（在这一点上 edX 支持三种不同的类别：家庭作业，实验或考试），我们将不同的得分分为不同等级（优秀，及格，不及格，未完成）。正如我们在图 5 中看

到那样，信息通过一个简单明了的方式呈现，使教师可以方便地分析结果。如果学生们在某个特定的章节遇到问题，教师可以很快的察觉到。

例如，通过这些图表，教师可以发现，平时的作业成绩通常比期中考试成绩高，这是正常的。但通过观察期末考试的结果我们可以看到，期末考试的成绩比平时的作业成绩还要高，虽然参加期末考试的人数要比之前低得多。而参与实验的人数则更加的少。

这些可视化提供的信息反馈对于课程未来的修改可能是非常重要的。有时候教师可能会需要更具体的信息，因此我们为有些图表提供了一些互动功能，通过点击或鼠标悬停提示，或者按类别细分。教师可以通过点击图表上的元素来访问有关每种类型数据的更详细的信息。图 6 是根据作业不同子小节分类构成的结果。

通过单独查看各个小节的统计，教师可以很容易判断是否有某个小节特别困难。如果有必要的话，教师可以单独对各个小节进行修改，让其难度满足要求。像"HW2" 和"HW5"，其中大部分学生都获得了非常高的分数或非常低的分数，这种情况表明，小节的设计可能有一些问题。通过这种方式，我们的教师可以很容易注意到这些问题。此外，在某些情况下教师可以检查学生是否最终会挂科，或者中途退出。

不同章节和活动的时间分布

第二个例子在图 7，这是一个显示同一课程中的每个章节学生所花费的时间。这是一个为教师设计的可视化，代表所有参加课程的学生花费的总时间。这可以让教师容易发现最耗时的章节，帮助他们来平衡不同章节的任务分配。该图表是一个百分比随第一章到最后一章逐渐降低的分布。这是因为 MOOCs 平台上的课程都有着很高的中途退出率，所以很多学生最后都没有完成课程。我们同样为这个图标增加了交互功能，通过点击图表上的任何章节，例如对应于第一章的蓝色区域，图标会变成图 8 那样。图 8 是被选择章节所花的时间分布图，根据活动的不同类型图表被分成不同的区域，例如有评分的活动（出现在蓝色）和没有意义活动（这将显示为红色）如视频，理论解释，教材阅读。而章节其他的部分（这是灰色的部分）保持不变。

这些可视化是针对整个班级而言的，对于个人信息的可视化我们会显示一些不同的时间行为分布。你可以找到中途退出的学生，也可以找到每章都投入

很多时间最后完成课程的学生，这些学生的行为大多是又学习动机和课程内容的复杂性决定的。还有一类学生，他们只阅读自己感兴趣的章节，对于其他章节则完全不管，这类学生没有完成整个课程，他们只完成其中的一部分。在任何情况下，教师都应该在下结论之前尽可能多的考虑这些图表提供的信息。

当测量学生在每个章节的时间分布时，有一个问题是我们如何判断用户是否有一段时间不活动。有时候，并不是十分容易来判断学生是在专心在阅读教材，还是在做其他的事情。虽然有一些 API 可以检测用户是否离开页面，或者离开键盘，但是要检测用户是否在观看教材和观看视频是十分困难的。这些类型的 API 可以减少统计的误差。也许，针对学生不活动的时间设置一个合理的阈值，是限制这种误差的一个方法。

A.6 结论和未来工作

MOOC 在过去的几年已经获得了很多成果。然而，大多数的 MOOC 平台还没有足够强大的学习分析功能，在这种情况下，教师和学生要从原始数据中获取有意义的信息就变得非常困难。当前，edX 在学习分析方面的支持也是不够的。在本文中，我们提出了实现 edX 的学习分析扩展的第一个步骤。我们同时也提供了一些技术问题的指导，总结和经验教训，争取在 open edX 上实现新的学习分析组件以及相关的可视化效果。在 edX 上的开发是非常具有挑战性的，因为这是一个结合多种技术，变化非常迅速的平台。

至于未来短期内的工作，我们的目标是定义和计算一些有趣的更高一级的指标，以及对应于这些指标的可视化效果。我们将参考其他著作中谈到的指标，同时结合我们自己的想法来决定应该使用哪些指标和可视化方案。

我们仍在努力开发新的可视化方案，改善 Open edX 平台上的教师和学生的使用体验。在将来，我们也想获得教师和学生使用这些工具的反馈，以评估我们的扩展。此外，我们要利用我们学习分析扩展提供的数据进行统计分析。最后，我们想实现一个推荐系统，可以防止学生中途退出同时鼓励良好的行为学习。