**Hadoop大数据处理行为痕迹记录的创新应用**

作者：杨秉学

**摘要**：随着计算机和网络应用的广泛深入，网络教学成为教育领域一个重要的组成部分，针对当前网络视频存在老师与学生之间交互性、实时性、反馈客观性等方面的缺陷，提出一种基于视频时间轴的用户播放行为监测H5播放器，实现了对学生观看视频全过程的行为监控、记录和反馈，客观地分析出该视频的重难点所在、学生的掌握情况等信息，辅助教学双方。

**关键词**：大数据、网络视频教学，时间轴，用户行为监测与反馈

一、研究背景与意义

随着互联网的发展，互联网越来越贴近人们的生活，人们的衣食住行，娱乐与学习等方方面面都有着互联网的身影。根据相关数据表明，人们每月花在上网的时间高达350亿小时，产生了庞大的互联网行为痕迹数据。依据国务院印发的《“十三五”国家信息化规划》，我国提出了大数据战略的重大决策，开启了信息化发展的新征程。教育部在2018年4月发布的《教育信息化行动计划2.0》也明确提出了促进信息技术在教育领域的广泛应用，推动教育的改革和发展，培养适应信息社会要求的创新人才以及促进教育现代化的目标。在社会进入快节奏的时代背景下，人们总是想方设法地用最高效的方式，获取最多的信息，而在教育领域里，视频学习的方式又逐渐成为了网络学习的主流。但由于视频学习方式本身的局限性，很多学生在通过视频自学的过程中并不了解课程的难点和重点，导致虽然看视频学了很长时间，效果却并不理想。而进行视频教学的老师，也无法了解学生们掌握的情况。因此，为了顺应时代的潮流、响应党和国家的号召、有效处理用户产生的大量行为痕迹数据，应用Hadoop大数据处理用户行为痕迹记录并将其以特定方式反馈给用户是大势所趋，网络教学持续不断地蓬勃发展，此项目紧抓机遇，针对教学情况反馈方面的缺陷，基于Hadoop大数据技术，将从网络教学视频中所获取的用户行为痕迹记录进行深刻地分析与处理，例如存储并处理用户的倍速播放的片段与次数以及来回播放的片段与次数等，将所获数据进行进一步的统计分析，应用Hadoop大数据技术全面客观的反映教学情况。鉴于时下HTML5的流行，为了解决这样的问题，我们提出用HTML5开发一个通过监测视频时间轴变化，来反映用户播放行为的播放器，通过抓取视频回放、前进的长度及次数、当前播放的速度等等数据，记录单个用户再视频学习中的播放行为，再使用大数据的相关技术将大量用户的数据加以统计分析，最后将统计分析结果在播放页面中以可视化的曲线、图表方式显示出来，以反映教学视频的重点、难点。对发展大数据的应用、推进社会信息技术的变革具有深刻意义。

在互联网迅猛发展的背景之下，授课打破了传统教学在时空上的限制，网络在线学习方式使得广大学生享受了丰富的教育资源，不仅实现了教育资源的共享，随时、随地皆可进行的远程教学模式也极大地便捷了教学双方，以网络课件、慕课(MOOC/MOOCs)等为代表的在线学习形式已成为协助教师教学、辅助学生自学的重要手段。然而由于网络教学是基于现代高科技的教育技术，毫无疑问会受到相关信息技术局限的限制，无法完全达到预期的教学效果。就目前网络教学的状况而言，在师生交流方面仍有极大的发展空间，尽管有弹幕，提问区域等方式促进教学互动，但这种交互方式学生的参与度并不高，也无完全地将问题表达，并且回复也难以做到全面与系统。教师对于讲义中的重点、难点以及学生的困惑难以准确地把握，使得传统教学中教师那种言传身教以及师生间的默契在网络教学平台中无法很好的实现，这是网络教学相对于传统教学的严重缺陷。

然而，人有悲欢离合，月有阴晴圆缺，事物总有两面性，在众多优点的背后，目前网课模式暴露出了下述的缺陷：

* 交互性不佳，教学双方均不能对自己的授课方式或学习进度给予客观的了解与反思。

在线学习网站的师生交流评论平台功能单一简单，加上积极响应的人不多，而且学习上出现的问题在讨论时往往不是三言两语、一言以蔽之就可以解释清楚的，这种简单的交互手段暴露出老师与学生之间交流不及时，回复个人化、片面化等等问题。使得教师无法获知学生的整体学习情况与自身的授课的薄弱环节，也使得学生难以把握整个科目的重点、难点。

* 实时性

针对网课中不同于传统授课方式，学生只能但方面的接收视频信息，而不能与老师进行一问一答的教学模式，这直接导致老师无法对学生的问题进行实时性的答复，并根据学生的反馈及时的对课程做出改进方案。

* 刷课

学生为了完成学校安排的任务，而远程授课的方式就可能存在造成学生刷课等不良现象的隐患，这些用户行为仅仅通过是否观看视频是不能全面地反映学生的知识掌握情况的。

为解决网络教学在交互性、实时性、反馈客观性等方面的缺陷，我们提出一种基于视频时间轴监测的用户播放行为监测播放器，对于多用户观看的单个视频，以帧为单位，记录学生观看视频时行为痕迹，例如视频的回放片段长度及次数、倍速播放长度等。通过对这些痕迹数据的统计分析，形成所有学生在观看教学视频时的总体行为记录，展示在视频教学网页上，以数字化的手段全面、公正地反馈教学信息，方便师生获得该视频教学中的难点和重点。教师可以明确掌握学生的学习情况，合理调节自己的教学进度，也可直观地发现自己教学的知识遗漏点、阐述不清等问题，及时补正，优化教学课件。学生可以对教学中重点难点的出现时间段有了一个基本的掌握，在此时间段会有集中注意力，认真听讲的意识，也可通过自己的观看情况对比大众的观看情况，反思个人的学习情况，及时发现自己知识储备的盲点和薄弱点。

依据此大数据的处理分析，能客观并且直观地发现教学中的难度较大与有待改进的环节，将学习视频中存在的问题全面且客观地反馈给教师，便于教师改进优化教学方案，提升教学质量，视频学习的优势得以最大化，而劣势则被弥补。其次，学生们可以清楚地看到大家所公认的重难点所在位置，有针对性，能够集中注意力地获取知识点，更了解自己知识的薄弱环节，便于洞察知识中的漏洞，清楚学习情况，做出科学合理的学习计划。老师们则可以及时掌握学生们有困难的地方，调整自己的教学内容和方式，为大家答疑解惑。

二、系统架构和算法设计

1. 语言及平台介绍

1)开发平台：

(1)Hadoop大数据用户行为数据处理

Hadoop和VMware虚拟平台。

VMware虚拟平台用于提供多个节点的，以实现部署hadoop以及实现分布式数据处理的需求。

Slave5 Slave6

Vmware

DataNode DataNode

Slave2 Slave3

Vmware

DataNode DataNode

Master与Slave通信以Slave2为例

：任务调度

：结果反馈

：心跳机制

：NameNode与SecondaryNameNode通讯

图1 系统节点分布架构图

该系统的Hadoop节点部署为1个master节点、6个slave节点，其中master与slave1、slave2、slave3以及slave4、slave5、slave6分别部署于三个不同的服务器上，并且slave2,slave3,slave5,slave6均为基于Linux系统虚拟化搭建的DataNode节点。

Hadoop平台用于对大数据进行分析与处理，将用户传进来的数据装入多个map里面，经过reduce处理得到分析的数据。

(2)H5播放器

软件开发平台：IntelliJ IDEA，Version: 2019-1；Eclipse，Version：2018-09 , H5用来做播放器以及对用户行为分析产生的结果在播放器上做出线条来标示。

2)开发语言：

1) Hadoop大数据用户行为数据处理

运用Javascript函数与ajax方法将前端获取的数据以JSON格式发送至服务器HDFS中。之后运用Java语言进行研究与开发，编写Job函数与MapReduce函数分布式处理用户的倍速播放的片段与次数以及前进、回退播放的片段起始、结束位置与次数等行为痕迹数据。

2) 播放器前端采用HTML5、CSS、JavaScript和Java语言设计开发，使用JavaScript用来捕获用户行为以及对播放器进行渲染，Html与css是用来制作播放器框架,后台服务器数据处理采用Java语言设计开发。

1. 数据处理与算法简述

反馈给前端

JSON

格式数据

HDFS

缓冲区

Shuffle阶段

Map阶段

Reduce阶段

图2 程序总体框架设计图

1）HTML播放器

采用SPring-mvc架构，应用HTML5语言制作网页播放器，使用JavaScript实现对用户视频观看行为的记录，将记录的数据转换为JSON格式，使用ajax方法发送请求将JSON文件写入HDFS中，之后使用java语言在Hadoop框架上，用MapReduce的方法对用户行为的数据进行计算、积累，获得每个视频以秒为单位的用户累计观看次数，并做成图表。当有用户观看该视频时，将最新的用户观看行为数据以曲线的形式显示在播放器进度条上，给当前用户作为参考。

考虑用户观看视频时可能有的暂停、继续、倍速播放、拖动进度条等行为，程序根据这些将用户观看视频的全过程划分为数个时间段，每个时间段有start、end、speed标记记录，分别代表该时间段对应的视频主时间轴的开始时间、结束时间（时间均以秒为单位记录）和倍速数值。其中，start、end、speed的初始值分别为0、0、1，表示用户从头观看视频，且当且播放进度为0，默认播放速度为1倍播放。

（1）数据获取——JS端发送数据

主要功能是事件请求及数据发送两个方面。

①调用video\_request.open函数来规定请求的类型、URL以及是否异步处理请求。使用 setRequestHeader() 来添加 HTTP 请求头。使用send(string)将请求发送到服务器。

②在当前页面获取request并调用getLine(video)，将数据文件放入本地或者HDFS中。

本文以鼠标点击“暂停”、“继续”事件的数据的获取过程为例做详细说明，其部分核心代码如下

注册侦听器：

<mx:LinkButton id="play" label="暂停" toggle="true" click="play\_clickHandler(event)" alpha="1" toolTip="播放/暂停" />

侦听函数：

protected function play\_clickHandler(event:MouseEvent):void

{

if(play.label== "播放")

{

videodisplay.play();

arr.add(videodisplay.currentTime);

}

else if(play.label == "暂停")

{

videodisplay.pause();

arr.add(videodisplay.currentTime);

}

}

具体来说，首先在mx标签中，为特定的对象增加事件响应函数。如示例中的“click="play\_clickHandler(event)"”，表示当用户单击该按钮时，将触发页面中的play\_clickHandler(event)函数。该函数，即侦听函数的具体定义以as3.0语句的形式嵌套在标签中，首先判断按钮类型，响应以控制视频暂停或继续播放；其次，通过访问视频文件舞台对象（视频初始化时由系统自动创建）的属性currentTime获取当前视频的帧数，并将其加入于Array中，数据初步获取完成。

进度条拖动、倍速播放等事件的侦听方式类似于上例，具体区别体现在侦听事件类型、数据赋值等方面。

2）start、end赋值（更改标题）

用户鼠标在时间轴上按下时，开启一个新的时间段，将此刻时间轴上的进度数据赋值给start；用户释放鼠标时，将此刻时间轴上的进度数据赋值给该时间段的end，一个完整的时间段记录完成。

1. speed赋值

speed数据记录主要通过全局变量current\_speed记录，初始值为1，在单个时间段结束前和end同时赋值。当检测到用户更改倍速播放的数值时，更改current\_speed的数值并给speed赋值，且本时间段记录结束，开启新时间段。

1. 数据整合

各个时间段数据的记录存放于数组中，当用户关闭播放器时，数据记录完成，将数组与该视频其他信息（播放视频URL等）打包传递给后续处理程序，由后续程序将这些记录转换为视频每帧的播放次数，记录到数据库。

程序框架图如下：

**后端**

**前端**

产生的相关数据

缓冲区

以曲线显示在网页上

数据存入HDFS

MapReduce

转化为JSON格式

图2 H5程序框架图

流程

1. 鼠标点击事件：暂停与继续

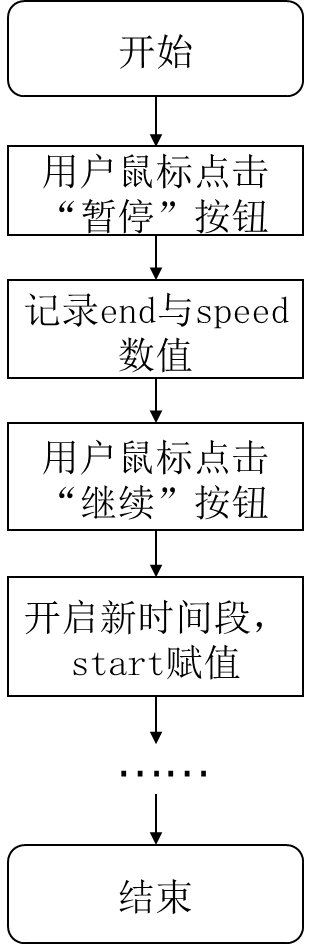


图 1 鼠标点击事件数据记录流程图

1. 部分按钮事件：倍速播放

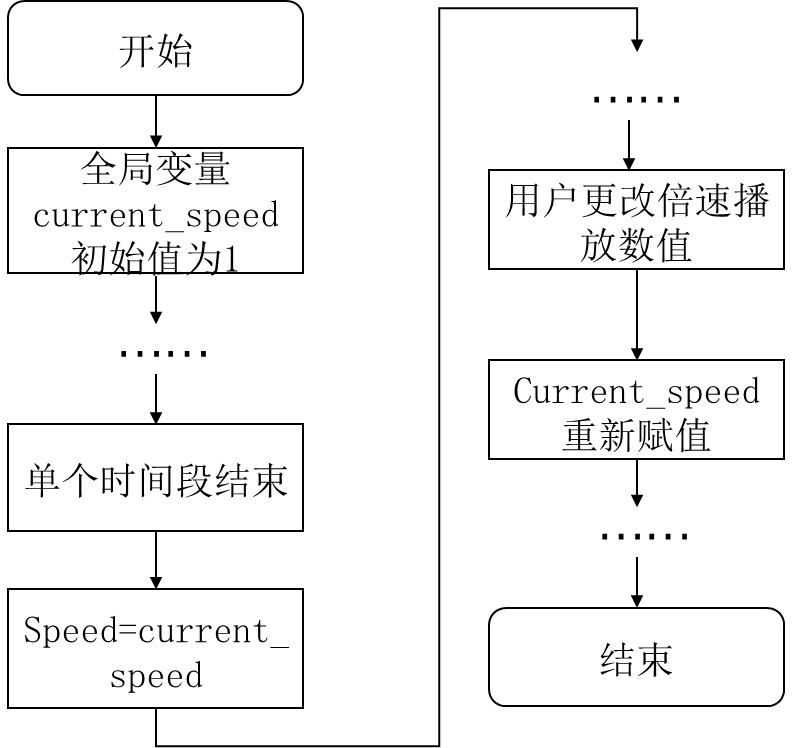


图 2 倍速播放记录流程图

3）鼠标拖动事件：拖动进度条

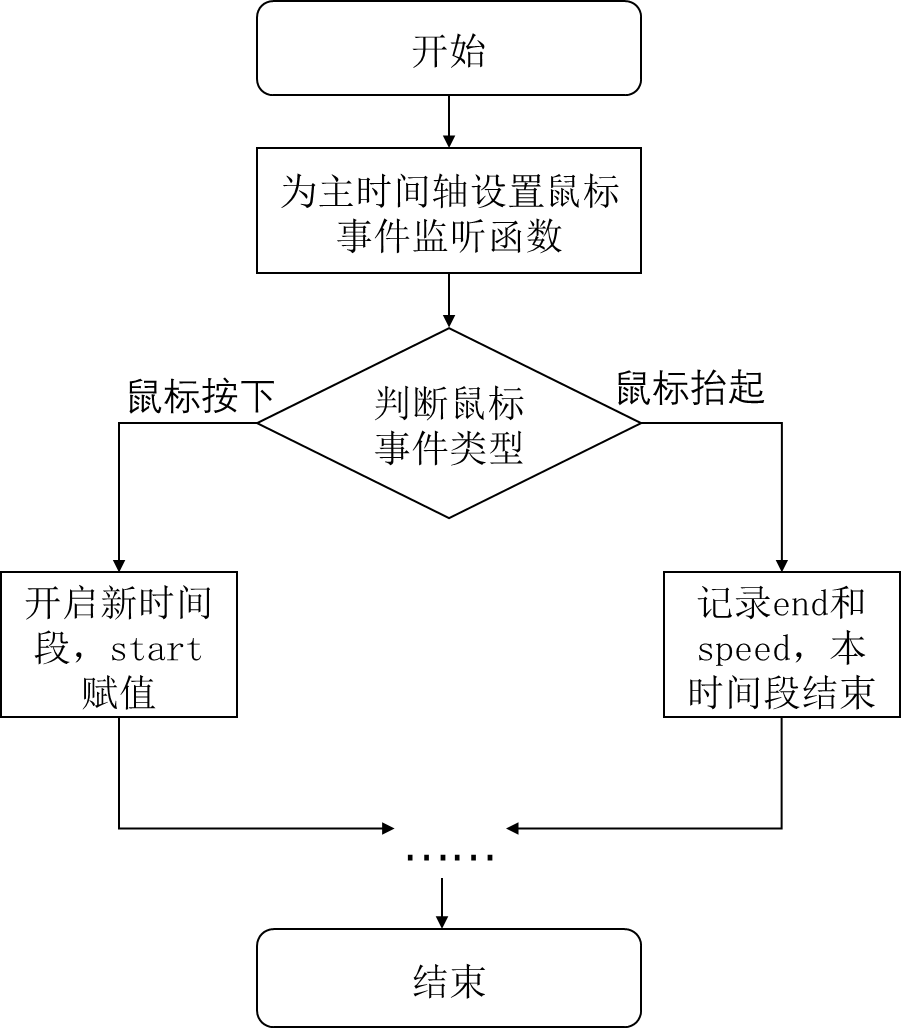


图 3 鼠标拖动事件数据记录流程图

1. Hadoop

Map阶段：各个节点将前端提交的每一条用户行为数据，按照多段起始时间及对应倍速，对数组进行加权计算，从而得到多个观看次数数组，数组的key值为视频URL，value值为某个用户观看该视频的过程中，对每秒视频的根据观看倍速进行加权的数值。

Shuffle阶段：MapReduce的关键环节，将key值相同的信息分到一类，并交给同一个Reducer处理，从而对信息进行整理和分类。

Reduce阶段：将key值相同的信息中的value值累加在一起，更新每个视频所有用户观看行为的总记录。

系统的程序算法设计图如下所示:

**Map**

**阶段**

URL1/ID1/start1/end1/speed1/start2/end2/speed2/......../start10/end10/speed10

URL1/ID2/start1/end1/speed1/start2/end2/speed2/......../start13/end13/speed13

URL2/ID3/start1/end1/speed1/start2/end2/speed2/......../start15/end15/speed15

........(此时数据格式为多个字符串)

URL1|ID1|1|1|1|1|2|2|2|2|2|2|1.5|1.5|1.5|1.5|..........|3|3|3|3|3|3|1|1|1|1|1|1|1|1|1|2|2|2|2|2|2|

URL1|ID2|2|2|2|2|2|2|2|3|3|3|3|3|3|3|4|4|4|1|1|1|1|...............|5|5|5|5|5|5|5|5|5|3|3|3|3|3|3|3|

URL2|ID3|1|1|1|1|1|1|1.5|1.5|1.5|2|2|2|2|2|2|2|2|1|1|1|1|........|0.5|0.5|0.5|2|2|2|2|2|2|2|4|4|4|

........(此时数据格式已转化为多个数组)

**Shuffle**

**阶段**

**URL1**|ID1|1|1|1|1|2|2|2|2|2|2|1.5|1.5|1.5|1.5|..........|3|3|3|3|3|3|1|1|1|1|1|1|1|1|1|2|2|2|2|2|2|

**URL1**|ID2|2|2|2|2|2|2|2|3|3|3|3|3|3|3|4|4|4|1|1|1|1|...............|5|5|5|5|5|5|5|5|5|3|3|3|3|3|3|3|

**URL2**|ID3|1|1|1|1|1|1|1.5|1.5|1.5|2|2|2|2|2|2|2|2|1|1|1|1|........|0.5|0.5|0.5|2|2|2|2|2|2|2|4|4|4|

........(Hadoop机制将key相同的信息归为一类)

**Reduce**

**阶段**

**URL1**|ID1|1|1|1|1|2|2|2|2|2|2|1.5|1.5|1.5|1.5|..........|3|3|3|3|3|3|1|1|1|1|1|1|1|1|1|2|2|2|2|2|2|

**+**

**URL1**|ID2|2|2|2|2|2|2|2|3|3|3|3|3|3|3|4|4|4|1|1|1|1|...............|5|5|5|5|5|5|5|5|5|3|3|3|3|3|3|3|

**=**

**URL1|3|3|3|3|3|4|4|4|4|5|5|4.5|4.5|4.5|5.5|5.5|4|4|4|4|4|...............|8|8|8|8|8|8|6|6|6|6|5|5|5|5|5|5|**

**URL2**|ID3|1|1|1|1|1|1|1.5|1.5|1.5|2|2|2|2|2|2|2|2|1|1|1|1|........|0.5|0.5|0.5|2|2|2|2|2|2|2|4|4|4|

+........(将key值相同的value累加在一起)

图2 Hadoop程序算法设计图

1. 框架设计
2. H5播放器

H5制作的播放器，主要框架，主要设计框架如图4所示。

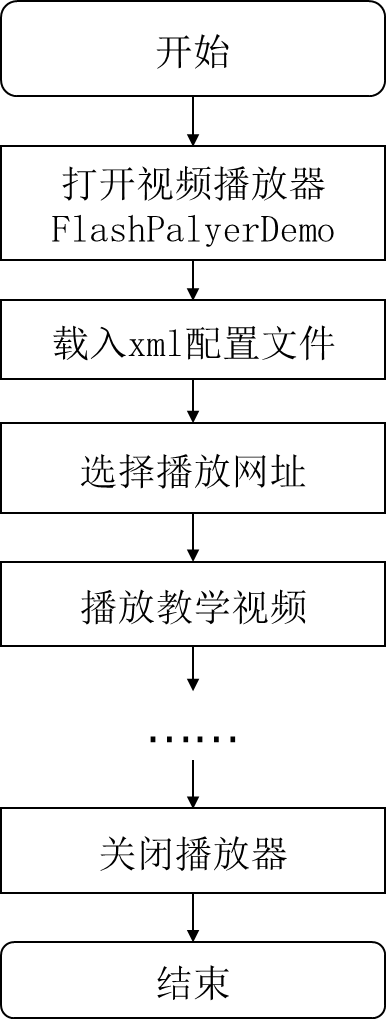


图 4 播放器实现框架

1. 用户行为记录

用户关闭播放器时，单个用户对于单个视频的观看全过程数据将以数组的形式传递给HDFS，HDFS以观看网址为表名，帧数（0~视频总长度）为列名，存储数据。

1. 数据通信
2. 播放器-页面通信

播放器中的数据信息（start、end、speed等）存储于Array类型的变量中，页面通过使用js语句调用flash中函数实现数值传递，部分核心代码如下所示。

播放器部分：

import flash.external.ExternalInterface;

function flashToJs(){

//数据信息作为函数返回值

return array;

}

//允许js调用flash中函数的申明

ExternalInterface.addCallback("callMe", flashToJs);

页面部分：

<body>

//flash组件说明

<object id="test" classid=

"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="">

</object>

</body>

<script type="text/javascript">

//js中调用flash中函数

document.getElementById("test").callMe();

</scritp>

1. 页面-Hadoop

使用ajax方法发送请求将JSON文件写入HDFS中，之后使用java语言在Hadoop框架上，用MapReduce的方法对用户行为的数据进行计算、积累，获得每个视频以秒为单位的用户累计观看次数，并做成图表。

3)Hadoop模块间通信

Master节点为NameNode节点，其将会给每一个slave节点布置数据块处理的指令，slave节点为DataNode节点，其为读、写请求提供服务以及按照NameNode的指令执行数据块的创建、删除与复制，并且当完成NameNode节点布置的数据处理任务之后，将存储数据处理结果并把结果反馈给NameNode节点。

NameNode节点与DataNode节点通过心跳机制进行通信，并以此监测各个节点是否正常运行。

NameNode与SecondaryNameNode之间的通信可合并NameNode的edit logs到fsimage文件中以防止当NameNode故障时导致fsimage文件的丢失。其通讯原理图为：

SecondaryNameNode

NameNode

fsimage

fsimage

定时查询NameNode上的edit logs

将edit logs改动更新至fsimage中

将fsimage拷贝回NameNode上

图 1 NameNode与SecondaryNameNode通讯原理图

1. 数据反馈与页面显示

主要功能是事件请求及数据接收两个方面。

1)创建XMLHttpRequest对象，用于在后台与服务器交换数据。并使用onreadystatechange()函数来检测XMLHttpRequest的状态。之后调用video\_request.open函数来规定请求的类型、URL 以及是否异步处理请求。最终使用send(string)将请求发送到服务器。

2）如果状态没有问题，则调用JSON.parse(this.responseText)来读取JSON中的文件。

因此，数据信息主要以折线图的形式显示于页面中。其中，以视频帧数为横轴，各帧平均播放次数为纵轴，通过曲线的峰谷值可大致判断该学习视频的重难点所在；曲线与横轴围成的区域颜色深浅表示在该教学网站中，该视频播放的热度高低，可供学生选课参考。

1. 创新点

播放器设计的创新点主要体现在：

* 以数字化的手段直观地记录用户的视频观看信息，信息反馈实时、可靠。
* 从大数据的角度，监测、存储和分析用户观看行为数据。
* 现已出现具有类似记录用户行为的html播放器，而本论文中的flash播放器具有更加良好的播放流畅性、稳定型和兼容性。