# 《短视频信息数据挖掘平台》

姓名：易晓博

学号：2018282110414

2018年12月01日

**文档修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **修改人** | **修改日期** | **AMD** | **修改说明** |
| V1.0 | 易晓博 | 2018.12.01 | A | 创建文档 |
| V1.1 | 易晓博 | 2019.01.09 | M | Docker部署分布式平台 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | 。 |

（A-添加，M-修改，D-删除）

目录

[一．项目背景 4](#_Toc534932817)

[二．需求分析 4](#_Toc534932818)

[三．系统架构 4](#_Toc534932819)

[3.1 平台初版架构 4](#_Toc534932820)

[3.2 架构更新 5](#_Toc534932821)

[3.2.1 Docker简介 6](#_Toc534932822)

[3.2.1 Docker改进架构 7](#_Toc534932823)

[四．阶段成果 7](#_Toc534932824)

[4.1 初步成果 7](#_Toc534932825)

[4.2 使用Docker改进后成果 10](#_Toc534932826)

[五．待改进完善点 12](#_Toc534932827)

[六．Github项目源码地址 12](#_Toc534932828)

一．项目背景

近年来国内短视频风潮兴起，成为国民进行消遣娱乐、消磨时间的重要途经，同时蕴含于其中商机和问题也逐渐暴露出来。例如：短视频中人气较高的主播成为网络明星，带来大量收入和流量，并且能引领话题风向；短视频中火爆的小吃美食，则为街边小吃美食的商铺带来机遇；在短视频快速发展的同时，其中也出现一些有违法律道德的信息，则需要依法封禁。

短视频信息数据挖掘主要针对于国内部分主流短视频平台，爬取短视频信息内容，对其数据包括但不限于标题、视频内容、作者、播放量、关注度、评论内容等进行数据挖掘；发现其中有价值的信息，例如可以引导网络流量的话题，民众喜爱的综艺、美食，将价值信息作为项目产品转售给第三方；同时如果发现有不良的违法信息，则及时向有关部门进行反映，协助监管部门进行监管。

二．需求分析

项目主要致力于有效挖掘出某个时间段内短视频信息，获取用户的话题风向、爱好兴趣等有价值信息。

三．系统架构

3.1 平台初版架构

本项目通过自动爬取国内抖音、快手、百度好看、优酷等短视频内容，进行分布式存储，并进行视频流数据分析。主要使用的框架及算法为OpenCV、Kafka、Spark、HDFS，具体架构如图3.1：

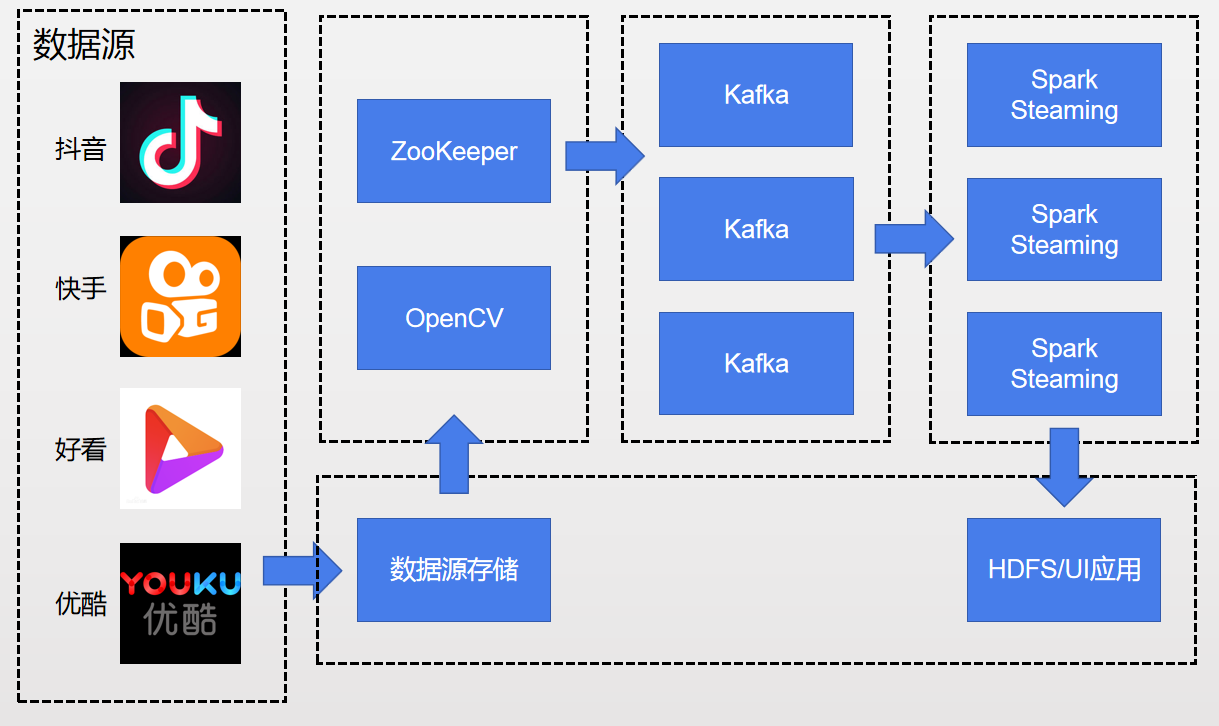


图3.1 短视频信息数据挖掘平台架构

OpenCV（Open Source Computer Vision Library）是一个基于BSD许可证开源的库。这个库使用C++编写，但是该库也有大量的Python、Java and MATLAB/OCTAVE的接口。OpenCV包含了数百个CV算法，能够用来处理和分析图片及视频文件。

Apache Kafka是一个分布式的流平台，它提供了一个发布和订阅流记录（streams of records）的系统。这些记录能够按照可容错的方式进行存储，消费者可以处理这些数据。

Apache Spark是一个快速、通用的集群计算系统。它提供了用于SQL和结构化数据处理的模块、用于机器学习的MLlib、用于图像处理的GraphX以及Spark Streaming。

项目使用OpenCV将视频流转换为帧数据，将这些数据交给ZooKeeper进行动态分配，由Kafka生成消息队列，Spark Streaming作为消费者从消息队列中不断取走视频流数据进行分析，最后将分析的结果展示到UI界面，或者在HDFS中存储起来。

3.2 架构更新

最近学习研究使用了Docker，根据Docker的特性考虑改进“短视频信息数据挖掘平台”项目中分布式平台搭建。因为在平常的大数据平台的开发运维工作中，发现现有的大数据集群部署方式复杂极其复杂，对于每一个大数据组件的安装都需要修改原有的配置文件信息，再修改过程中也很容易出错。当然市面上也有开源的和商用的集群管理组件如Ambari（HDP），CDH，但是开源的管理组件安装服务仍然容易出错并且错误原因不好查找，而商用的集群管理工具不能扩展，导致很多平台功能不能往下开发。但是我们平台的环境搭建特色就是一站式部署，减少中间手动的改动流程，让环境部署更轻松，更便捷。

# 3.2.1 Docker简介

Docker是使用Google公司推出的Go语言进行开发实现， 基于Linux内核的CGroup、Namespace以及AUFS技术，对进程进行封装隔离，属于操作系统层面的虚拟化技术。由于隔离的进程独立于宿主和其他隔离的进程，因此也被称为容器。与传统虚拟机和Docker容器技术比较，虚拟机不仅包括应用也包括操作系统。 操作系统一般要占用几个GB的磁盘空间，而Docker容器共用宿主的内核，只需按照应用和必要的类库，非常轻量。



图3.2 Docker简介

如果在测试服务器上使用Docker，会极大节省应景资源，简单企业成本。 Docker支持镜像构建和镜像部署，有类似Github的Docker hub的镜像开源仓库hub.docker.com, 这上面有100000多免费镜像资源，目标国内也有很大Docker镜像网站。使用Docker容器，只需获取镜像，一键安装即可。同时，也可以将常用的系统软件集成构建成镜像后部署到其他容器。支持测试环境和线上环境使用Docker容器技术进行软件的构建发布。

# 3.2.1 Docker改进架构

使用Docker之后，我们之前的软件架构也有所调整，并且加入了Hive、Hbase、Hue等组件，这是为了后续平台更好地应对非结构化数据而新加入的，同时利用Hue这一可视化的开源图形界面，可以极大的提高开发人员的运维效率。

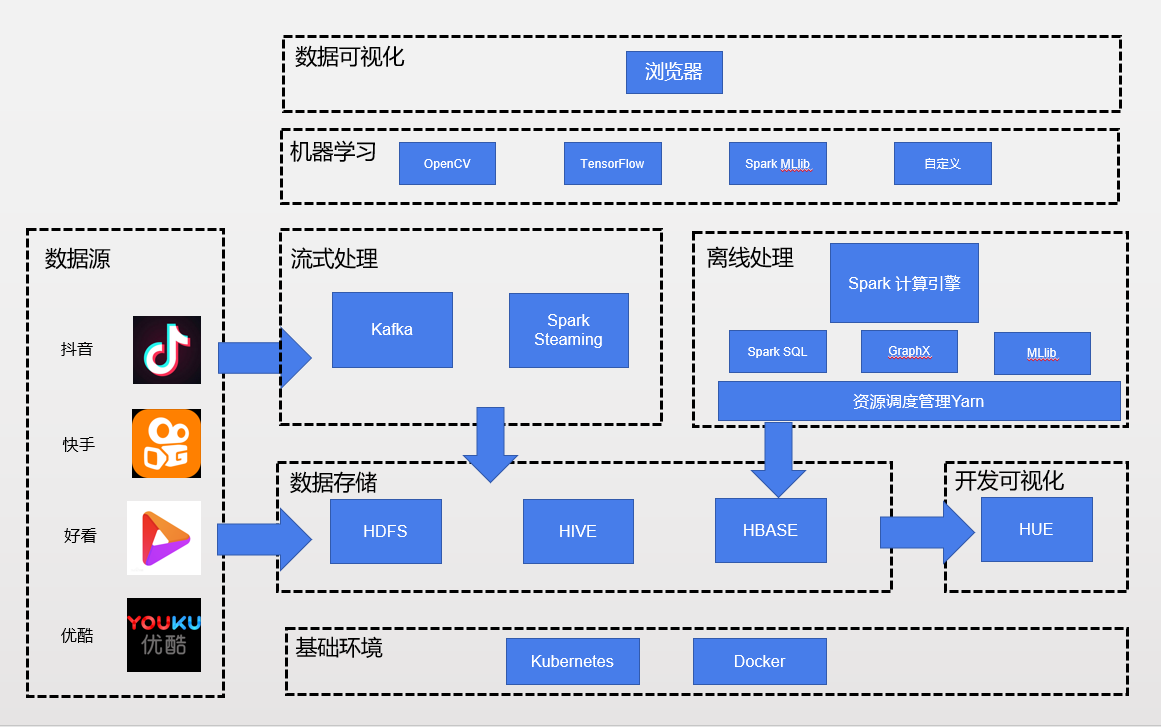


图3.3 使用Docker改进后平台架构

四．阶段成果

4.1 初步成果

1.视频爬虫处理

目前项目已经完成快手的爬虫处理，使用脚本从Fiddler批量下载数据包数据，从中解析出视频数据信息。包括视频发布者ID、视频播放量、视频关注度、评论、视频下载地址等等；爬虫Demo可以参考GetVideoInfo源码Demo，其中包含一个测试的数据包URL集以及下载完成待解析的视频。

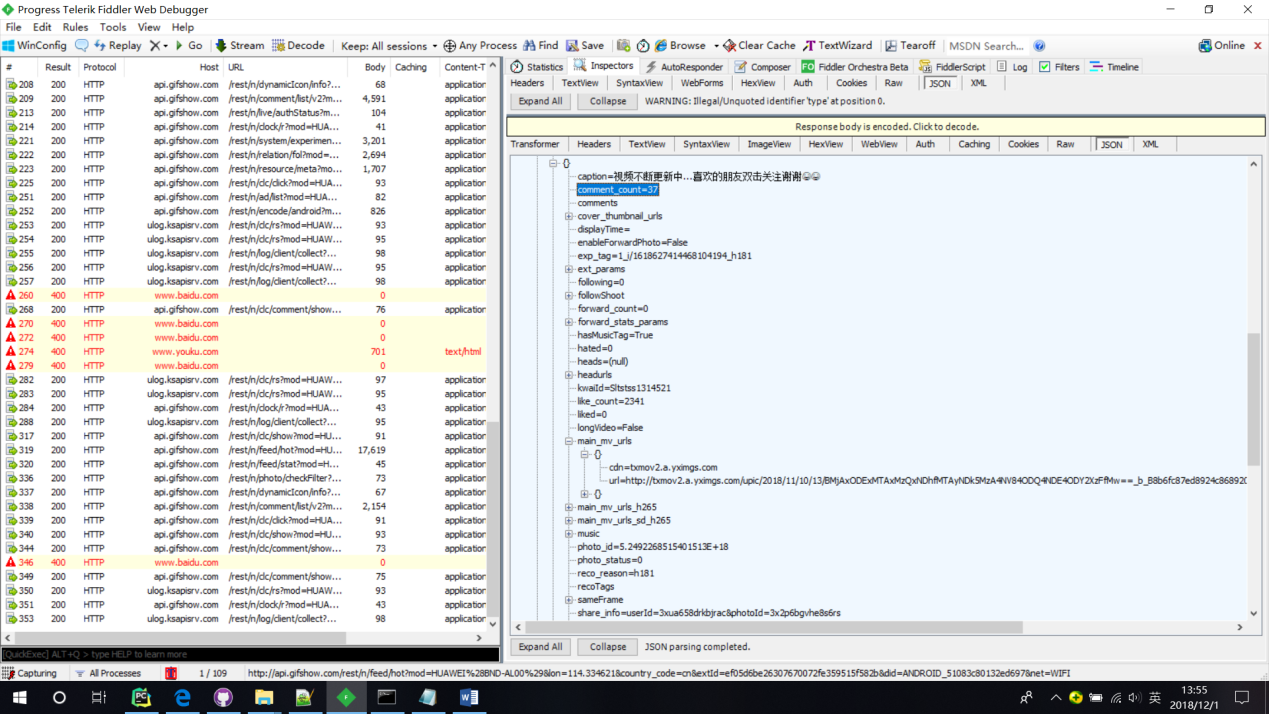


图4.1 Fiddler爬取视频URL

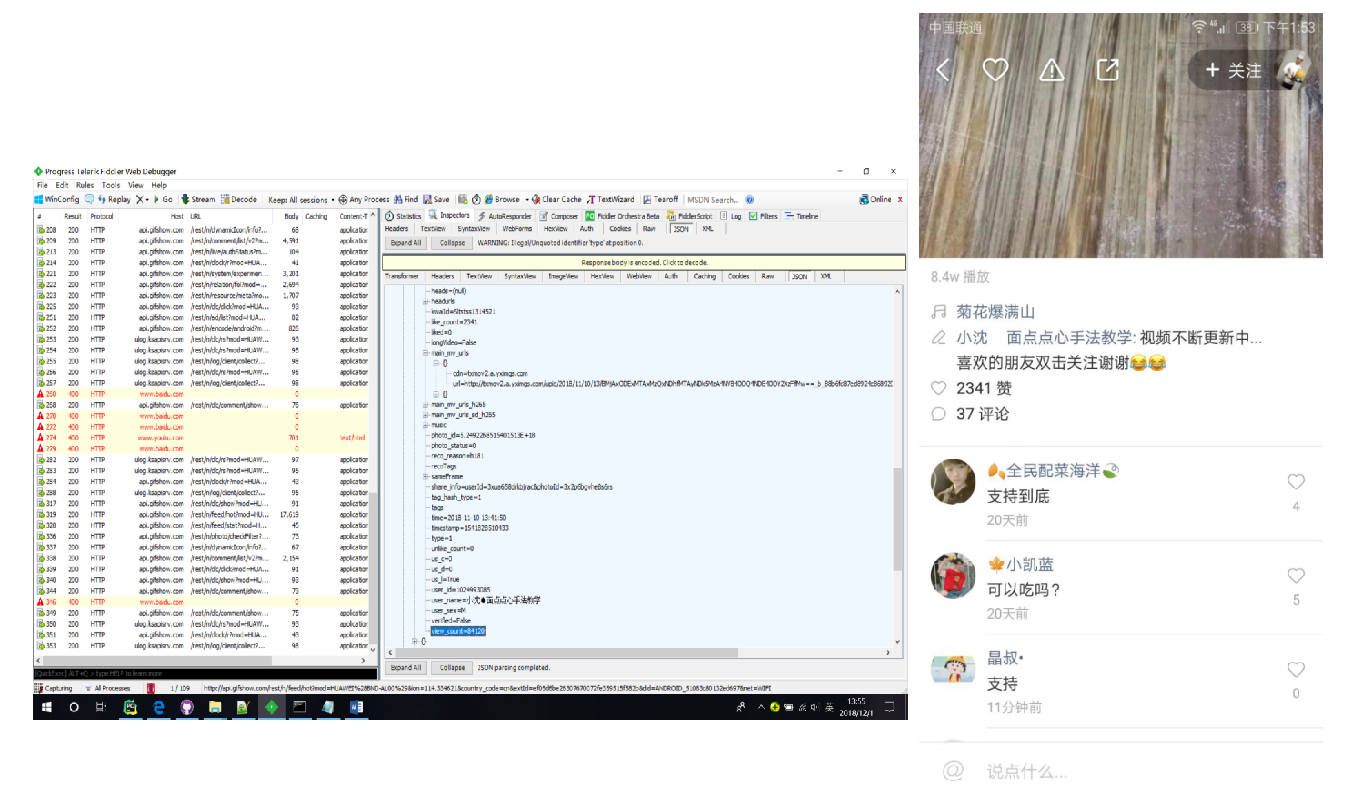


图4.2 Fiddler爬取信息与视频客户端显示信息一致

可以明显看到解包数据与在APP上看到的视频数据一致。并能正确从视频服务器端下载对应视频，留在处理解析。

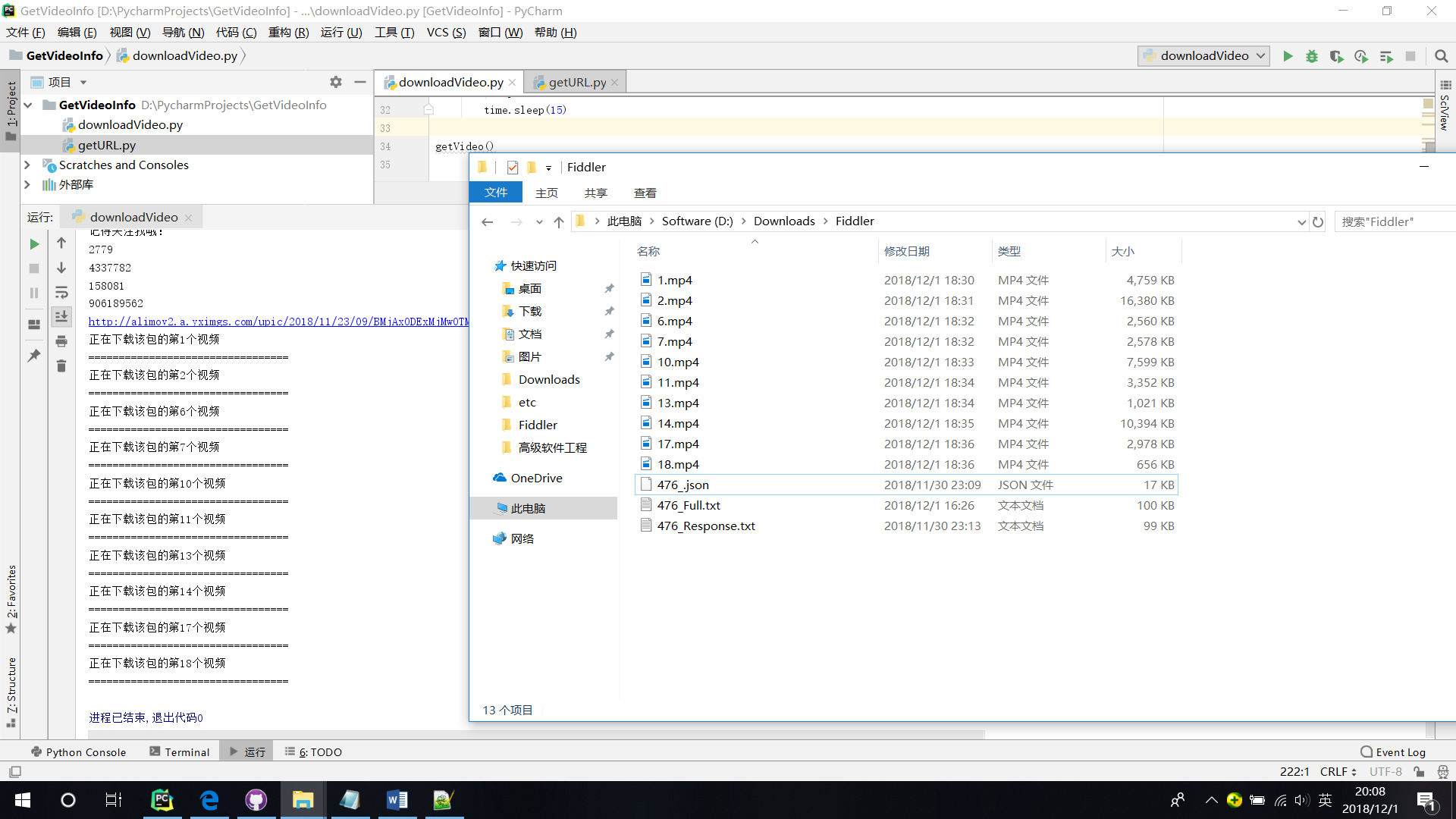


图4.3 服务器端从网络爬取视频

2. ZooKeeper+Kafka+Spark+HDFS分布式平台框架搭建

在本地虚拟机上测试消息队列和SparkStreaming消费者获取数据正常，并能正常完成部分数据分析任务，本地PC性能有限，目前测试3个Spark节点。

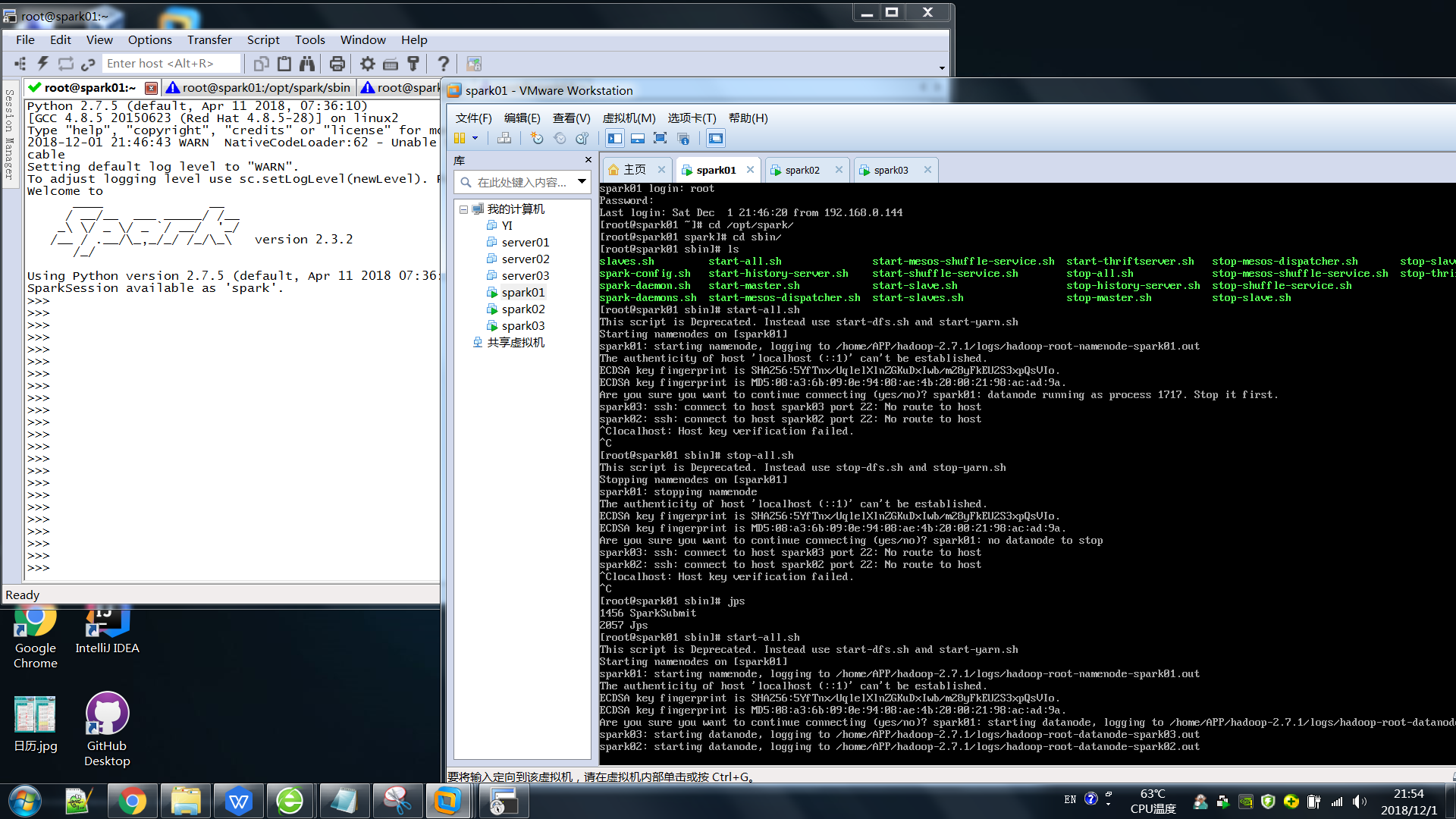


图4.4 Spark分布式集群搭建

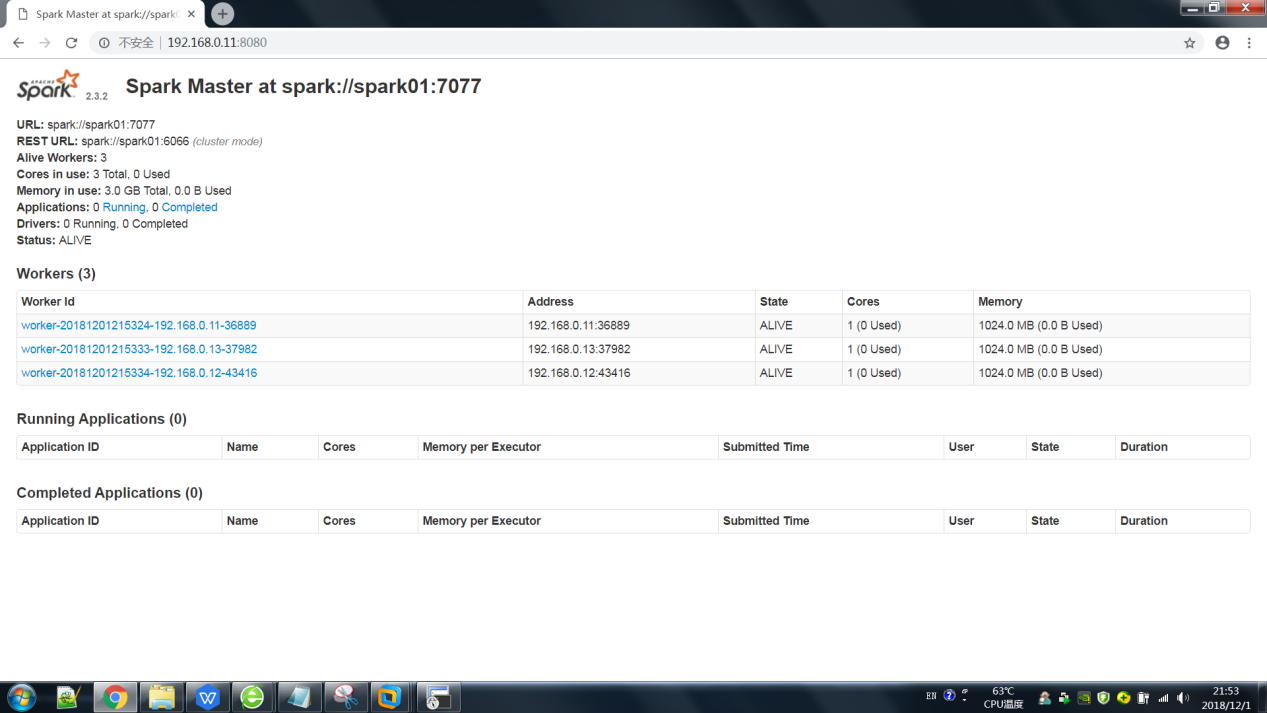


图4.5 Spark WebUI管理界面查看数据分析任务

Hadoop、ZooKeeper、Kafka、Spark分布式安装过程比较简单，项目后期会采用CDH进行集中管理，在此以上开源框架的配置与安装过程就不再过多赘述。

4.2 使用Docker改进后成果

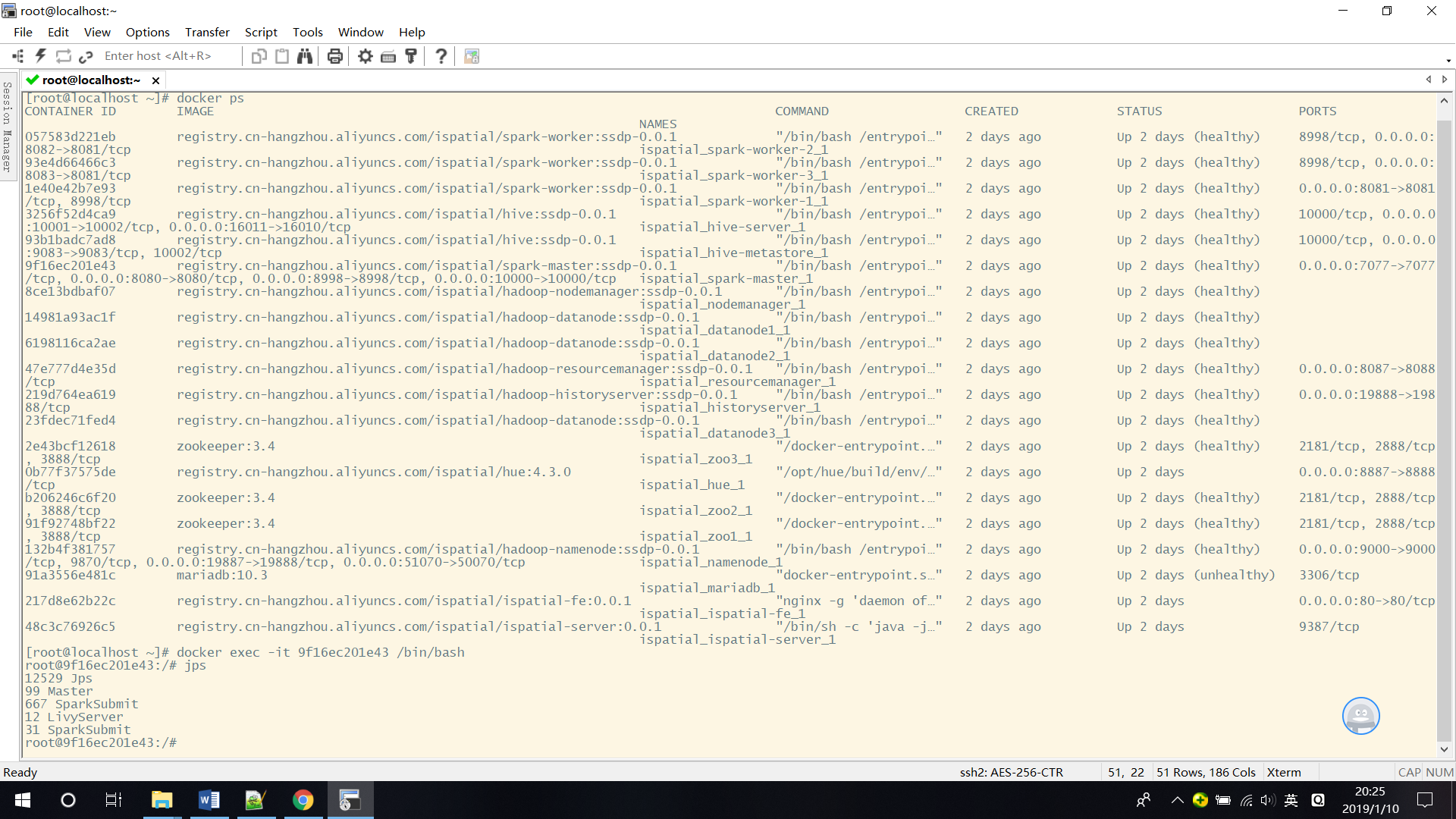
下面是Docker在虚拟机上运行的实例，后续平台搭建可以参考。从下图我们可以看到各个大数据框架组件已经正确配置，同时查看到Spark的Master节点在Docker镜像中正常运行。

图4.6 Docker配置大数据框架组件运行实例

同时我们部署了Hive数据仓库管理工具，方便后续平台对视频数据、爬虫数据的管理。一方面，我们可以利用Hue等可视化工具进行Hive数据仓库的运维工作；另一方面，我们利用SpringBoot来进行底层开发，为后续自主开发UI界面，来操作对数据增删改查等功能奠定基础。

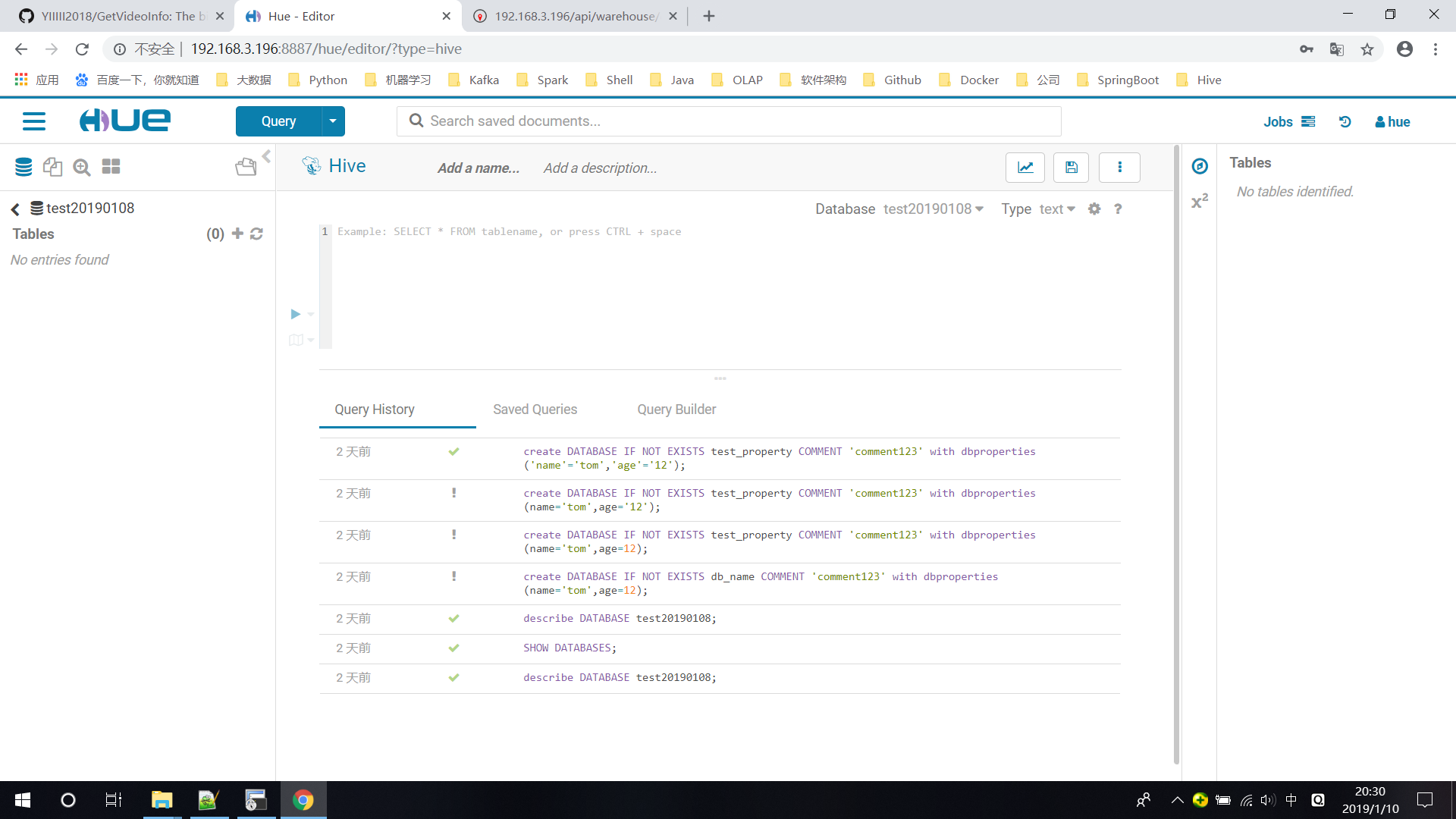


图4.7 Hue可视化管理数据库

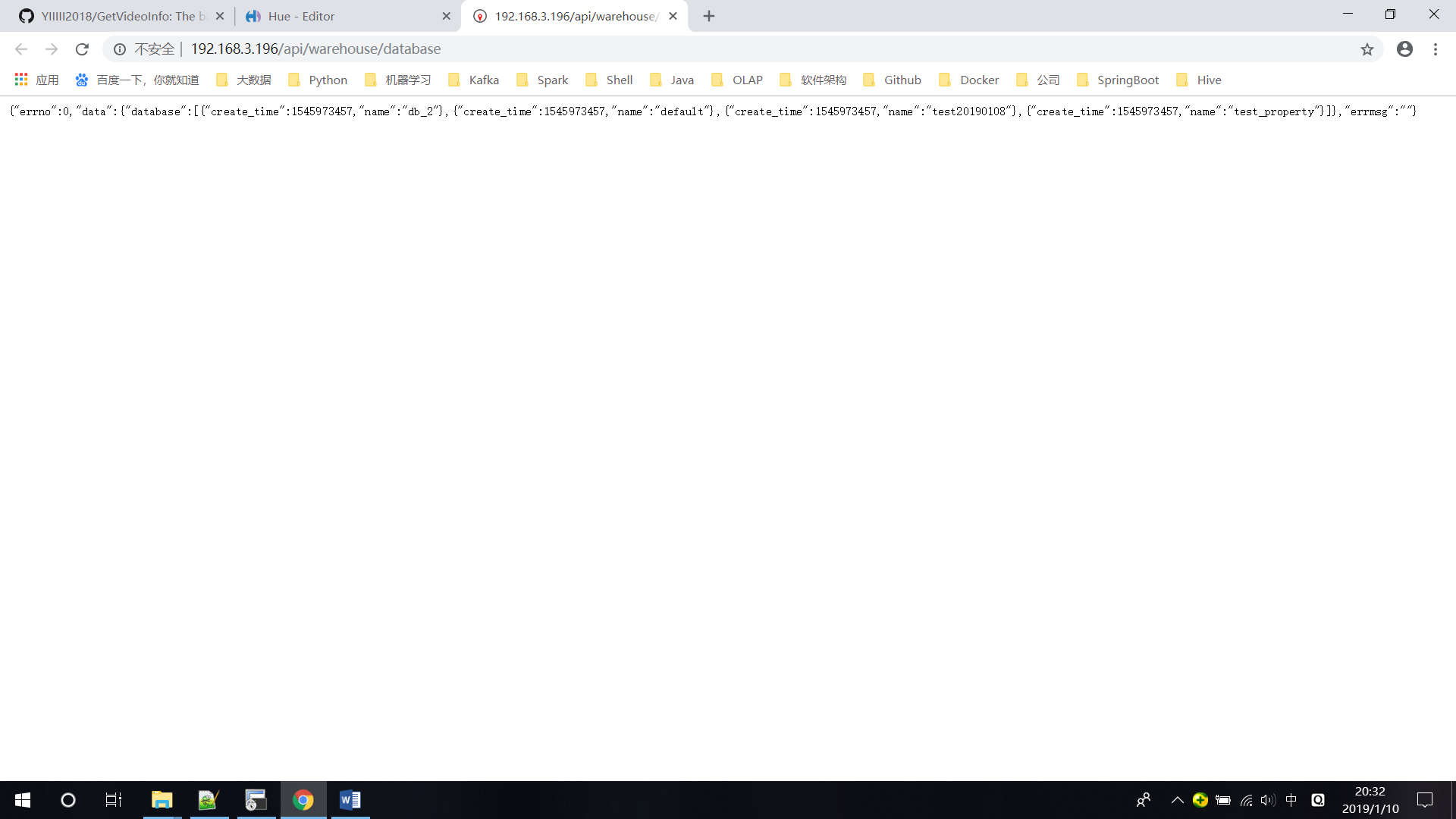


图4.8 使用SpringBoot成功获得返回的Json对象

五．待改进完善点

目前爬虫处理需要在手机端进行抓包，虽然抓取视频可以进行多线程处理，但是视频数据包的来源速度受限，一台手机仅一个数据包来源，且使用脚本从Fiddler解析数据包，属于半自动化的过程，希望能提高效率；

OpenCV目前工作缺少相应专业人员推进工作，下载的视频数据待处理解析；需要熟悉数理统计的人员，为清洗处理过的流数据进行建模分析，最好熟悉分类、回归、聚类算法，完成挖掘出有价值的视频信息的工作。

六．Github项目源码地址

https://github.com/YIIIII2018/GetVideoInfo.git