**基于大数据平台的日志分析平台**

1. **项目介绍**

Web日志由Web服务器产生，可能是Nginx, Apache, Tomcat等。Web日志包含着网站最重要的信息，通过日志分析，我们可以知道网站的访问量等多种信息。一般中型的网站，每天会产生1G以上Web日志文件，大型或超大型的网站，可能每小时就会产生10G的数据量。当数据量每天以10G、100G增长的时候，单机处理能力已经不能满足需求。对于日志的这种规模的数据，用Hadoop进行日志分析，是最适合不过的了。

项目成员：

组长：徐磊，算法开发

组员：颜宇辰，后端

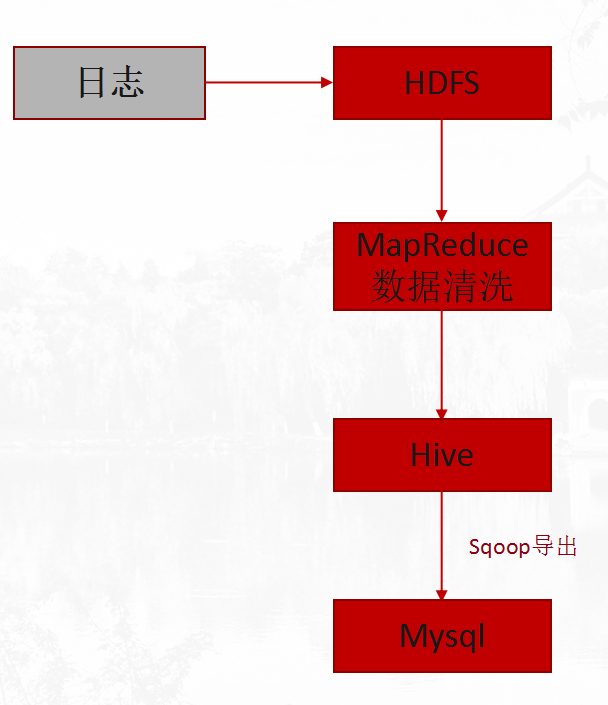
闫涵，算法开发

高垚鑫，前端

冉然，数据采集，环境搭建

链接：<https://github.com/anaana35/work>

1. **系统架构设计**



系统架构图

1. 把日志数据上传到HDFS中进行处理；

2. 使用MapReduce对HDFS中的原始数据进行清洗；

3. 使用Hive对清洗后的数据进行统计分析；

4. 使用Sqoop把Hive产生的统计结果导出到mysql中。

5. 前端界面进行展示

1. **环境搭建**

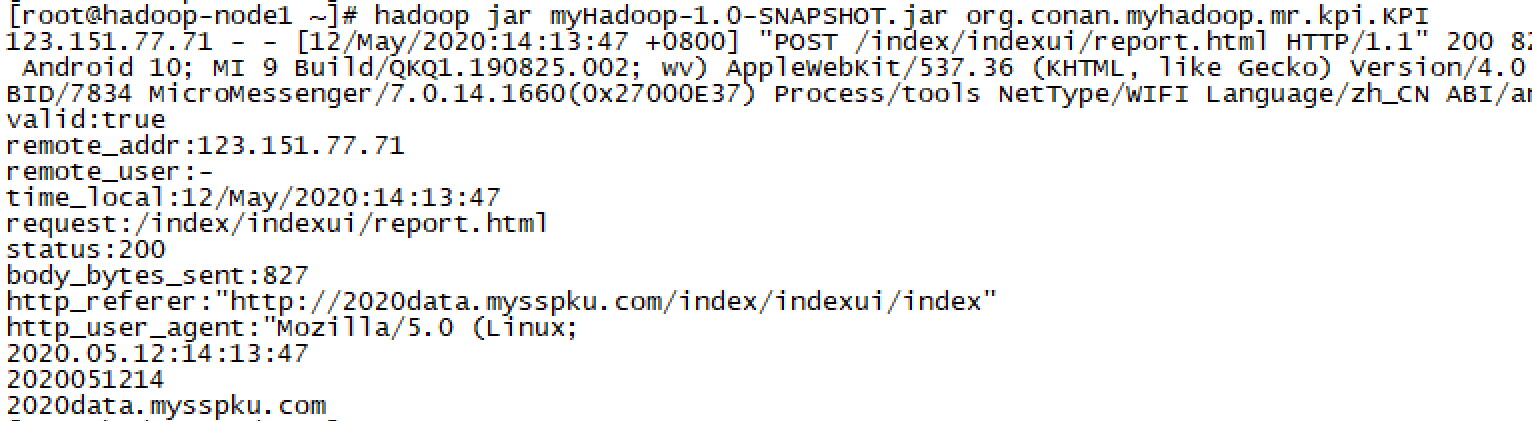
在华为云主机上（2核4G，CentOS7.4操作系统）安装部署三个节点的伪分布式hadoop大数据平台。其中，Node1节点为NameNode和DataNode，Node2节点和Node3节点为DataNode。在Node1节点上安装hive环境，在Mysql中创建hive数据库。

1. **数据采集**

本次实验的数据集为老师提供的疫情期间学生健康打卡系统的日志数据。该数据集为Nginx格式的日志，有接近150万行数据，无动态增长，采用离线分析的形式，将数据集上传到HDFS文件系统的input目录下。

1. **数据处理**

HDFS中的原生数据为不规整数据，该数据的格式还无法满足我们对数据处理的基本要求，需要对其进行预处理，转化为我们后面工作所需要的较为规整的数据。所以数据清洗，就是利用hadoop的map-reduce对数据进行基本的预处理，以方便我们后面的统计分析。



Web日志分析概述Web日志由Web服务器产生，可能是Nginx, Apache, Tomcat等。从Web日志中，我们可以获取网站每类页面的PV值（PageView，页面访问量）、独立IP数；稍微复杂一些的，可以计算得出用户所检索的关键词排行榜、用户停留时间最高的页面等；更复杂的，构建广告点击模型、分析用户行为特征等等。  
 在Web日志中，每条日志通常代表着用户的一次访问行为，例如下面就是一条nginx日志：

222.68.172.190 - - [18/Sep/2013:06:49:57 +0000] "GET /images/my.jpg HTTP/1.1" 200 19939  
"http://www.angularjs.cn/A00n" "Mozilla/5.0 (Windows NT 6.1)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.66 Safari/537.36"

拆解为以下8个变量  
remote\_addr: 记录客户端的ip地址, 222.68.172.190

remote\_user: 记录客户端用户名称, –

time\_local: 记录访问时间与时区, [18/Sep/2013:06:49:57 +0000]

request: 记录请求的url与http协议, “GET /images/my.jpg HTTP/1.1″

status: 记录请求状态,成功是200, 200

body\_bytes\_sent: 记录发送给客户端文件主体内容大小, 19939

http\_referer: 用来记录从那个页面链接访问过来的, “http://www.angularjs.cn/A00n”

http\_user\_agent: 记录客户浏览器的相关信息, “Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.66 Safari/537.36″

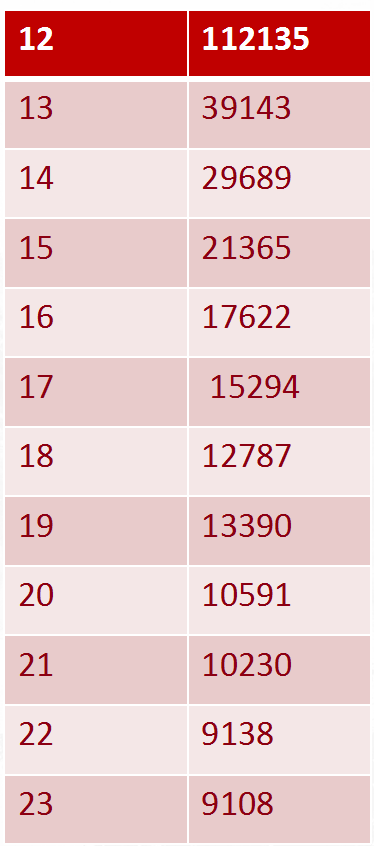
本次实验我们主要统计了三个信息：

1.独立IP信息

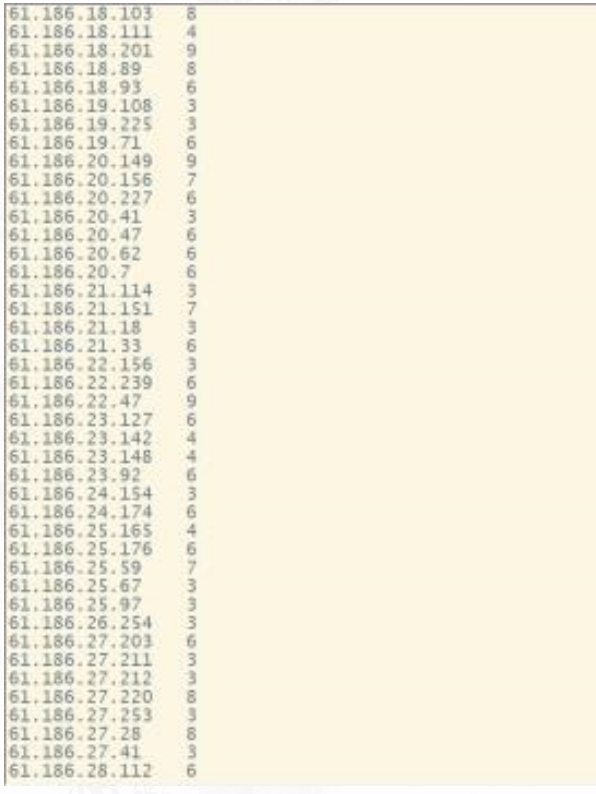
2.用户每小时PV的统计

3.用户的访问设备信息

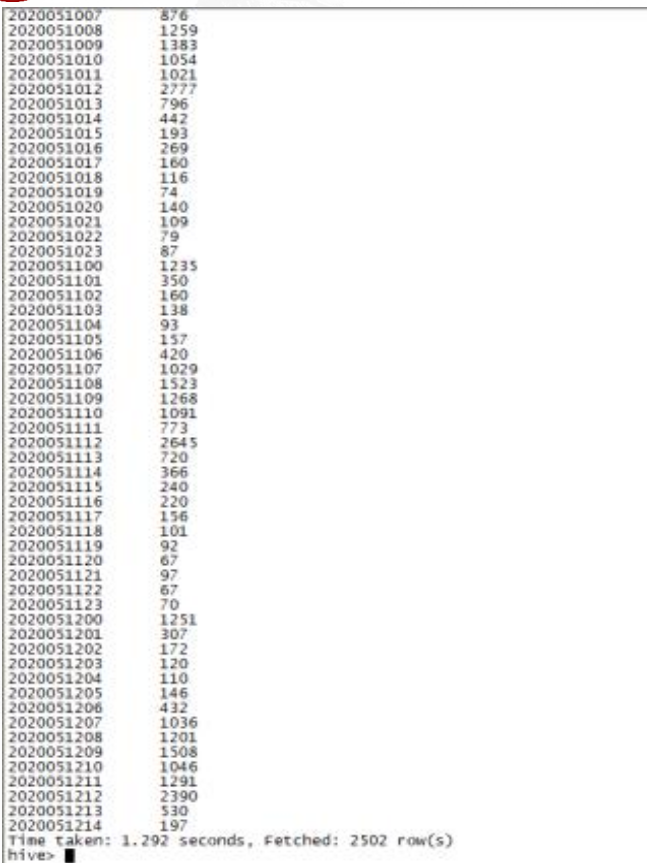
按时间的打卡人数统计：

独立IP信息：

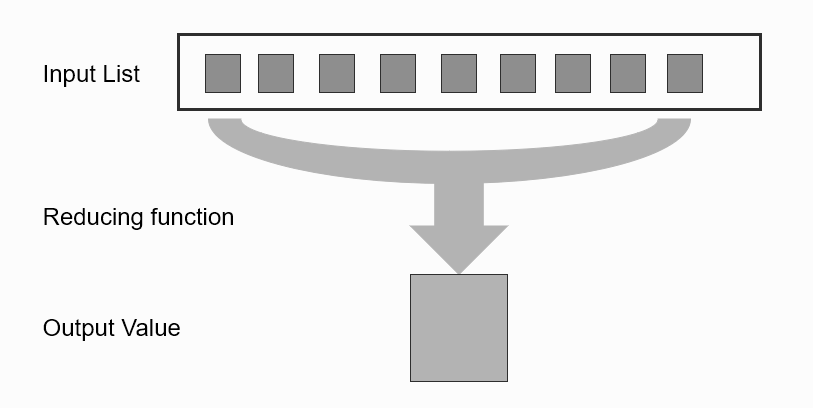


每小时访问量的统计：



1. **算法开发**

算法模型采用hadoop的并行算法。



MapReduce程序开发流程：

用Maven构建hadoop项目，首先对日志进行解析，将每条日志进行切分，找到有定义的变量。

PV(PageView): 页面访问量统计

Map过程{key:$request,value:1}

Reduce过程{key:$request,value:求和(sum)}

IP: 页面独立IP的访问量统计

Map: {key:$request,value:$remote\_addr}

Reduce: {key:$request,value:去重再求和(sum(unique))}

Time: 用户每小时PV的统计

Map: {key:$time\_local,value:1}

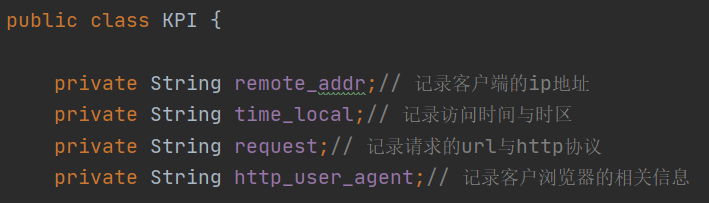
Reduce: {key:$time\_local,value:求和(sum)}

Browser: 用户的访问设备统计

Map: {key:$http\_user\_agent,value:1}

Reduce: {key:$http\_user\_agent,value:求和(sum)}

定义KPI类：



将日志行正确解析成KPI对象的属性，把解析过程单独封装成一个方法parse。把map方法，reduce方法单独写一个类实现，分别为KPIPV，KPIIP,KPITime，KPIBrower。

最后将项目打包，传入hadoop-node1节点运行，结果存入HDFS文件系统的output目录。

1. **数据库操作**

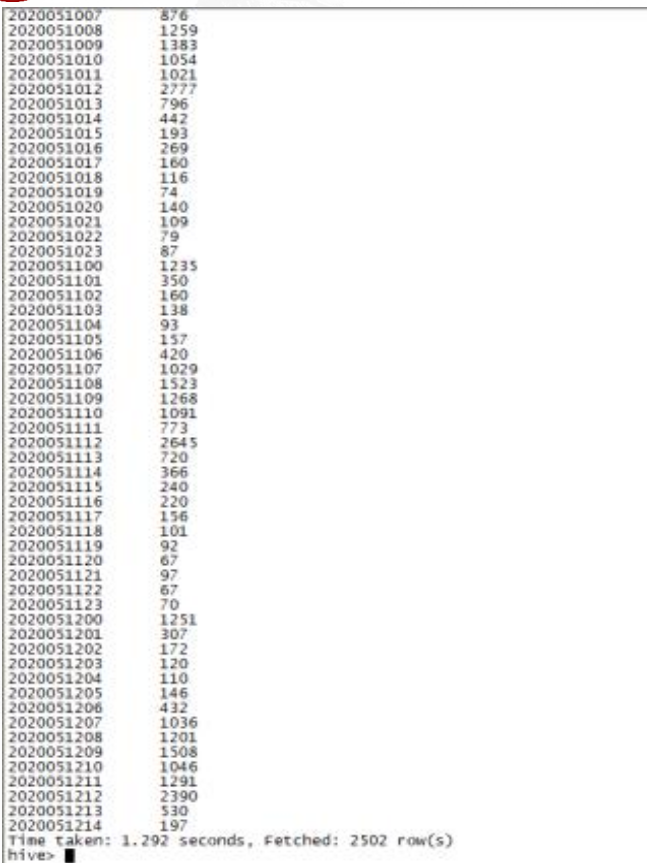
首先将MapReduce清洗后的数据存入hive，这部分可以参考张老师的教程。

之后，使用hive对数据进行进一步处理。

建立三个表，分别为Time（time int，num int）,Browser(browser string,num int)和IP（ip string, num int）.

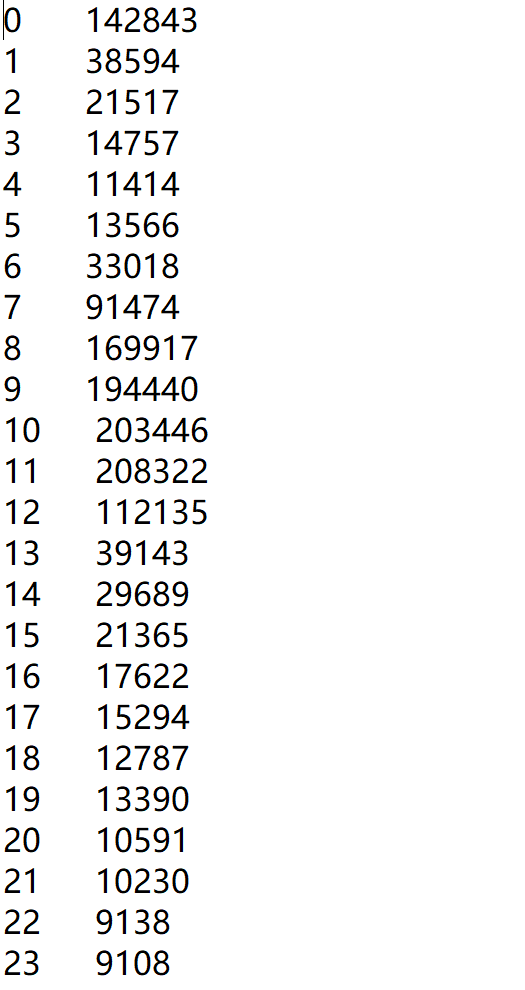
举例：统计各个时段的打卡人数

原始数据：



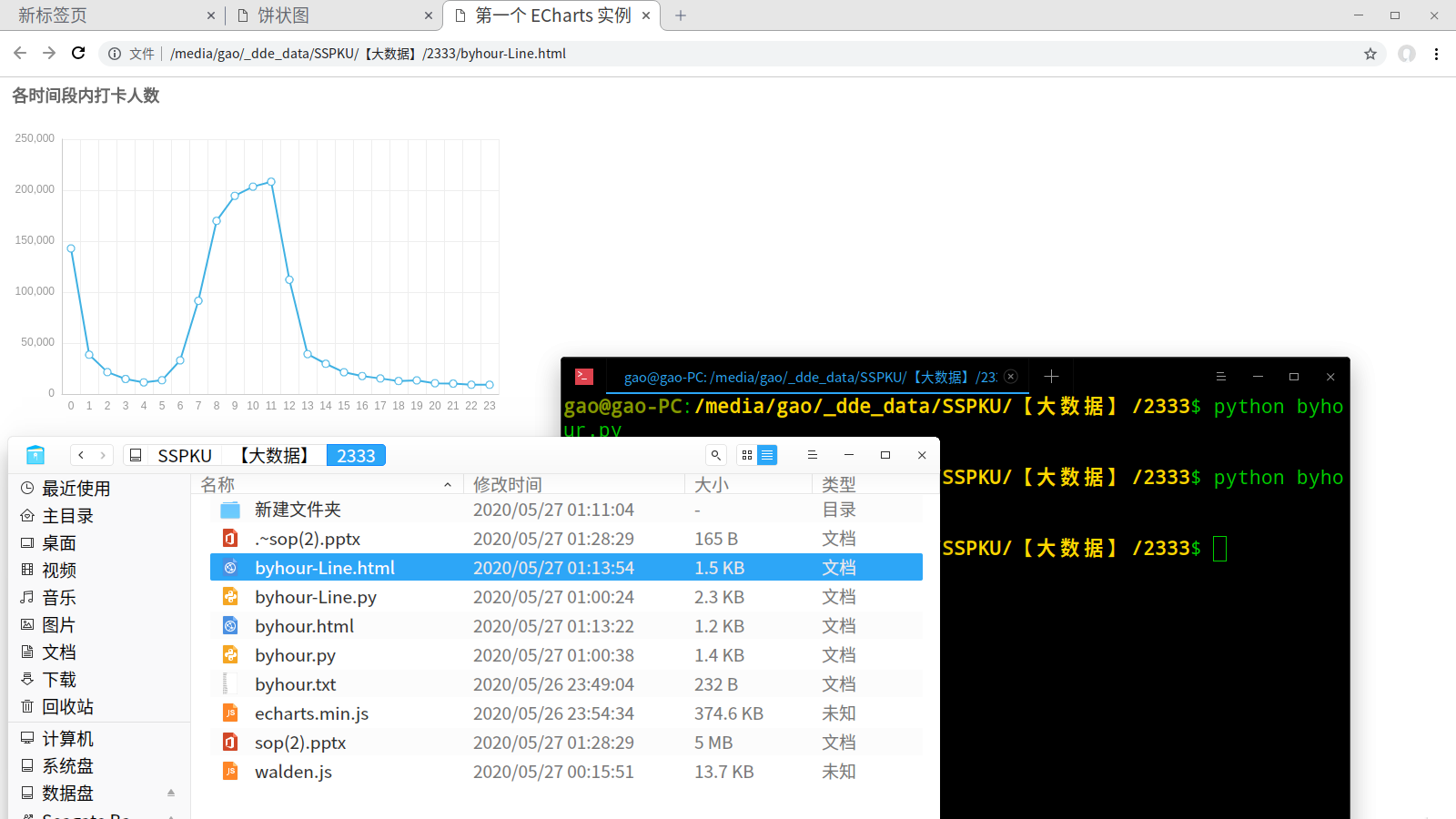
使用select time%10, sum(num) from Time group by time%10

获得：



1. **后台服务开发**

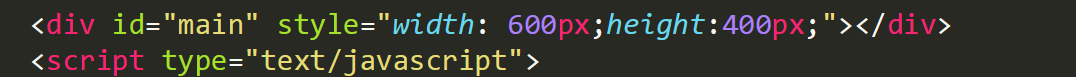
通过Python程序，将数据渲染成网页 .txt (raw data) ----(.py)---->.html (user-friendly webview)



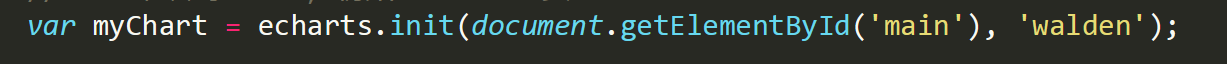
1. **前端开发**

前端部分使用JavaScript实现的开源可视化库ECharts,使得数据的改变驱动图表展现的改变。

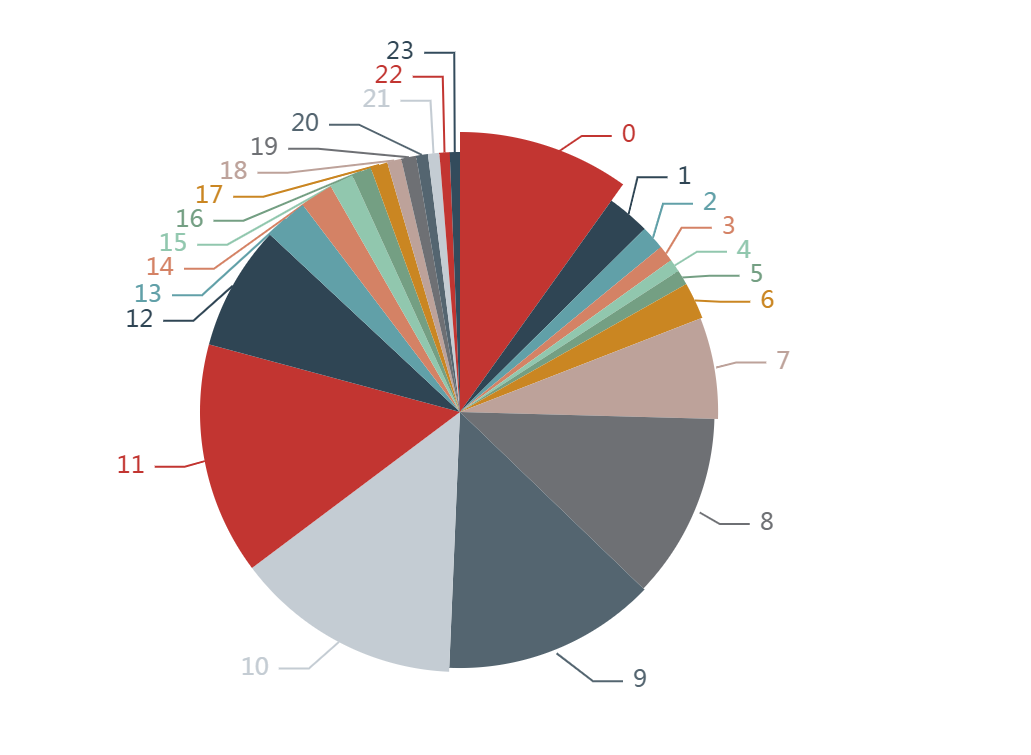
为 ECharts 准备一个具备高宽的 DOM 容器：



基于准备好的dom，初始化echarts实例：



前端界面显示的各时间段内打卡人数饼图显示：



各时间段内每小时打卡人数折线图，使用walden.js切换主题。

