****

分布式系统

期末项目

**指导教师：李江峰**

**小组成员：李倩**

**朱西宁**

**于陈峰**

**黎龙奇**

**刘雅骏**

**日期:** 2018/1

**一. 网络结构**

1.1 集群拓扑结构

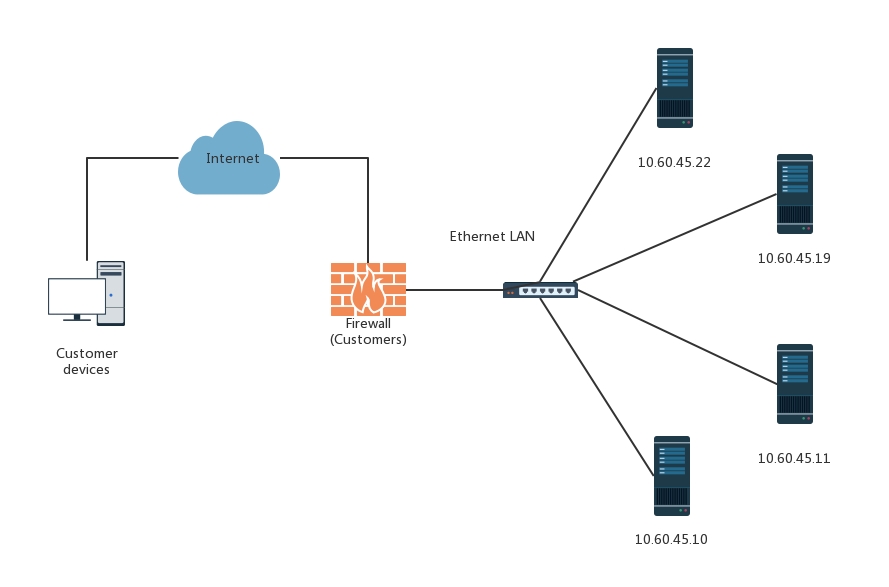
集群包括了4个节点，网络ip分别为10.60.45.10, 10.60.45.11, 10.60.45.19, 10.60.45.22。这4个节点处于同一局域网下，与同一台交换机相连。每个节点上都在端口8000上设置了web服务程序。用户可以通过访问web服务程序的方式来进行文件操作。

Web服务程序由基于python的Django框架实现。

1.2 通信

4个节点位于同一局域网下，操作系统都使用了Ubuntu 16.04，可以直接通过ssh或ftp的方式来进行互相访问。在搭建集群时，在每个节点上都设置了集群内文件操作专用账户，账户名和密码加密后存储在每个节点的同一文件路径下。

当节点需要向其他节点发出请求或者传送文件时，就从本地读取密码文件，然后与对应节点建立连接。

 网络通信由python第三方库paramiko实现。

1.3 Web服务程序

在这部分我们使用的是python的Django框架，Django是MVC框架中的一种，使用MVC来编写这个系统有一点好处是能够使科幻短和服务端相互分开，符合C/S架构。

Django中提供了开发网站经常用到的模块，常见的代码都为你写好了，通过减少重复的代码，Django 使你能够专注于 web 应用上有趣的关键性的东西。为了达到这个目标，Django 提供了通用Web开发模式的高度抽象，提供了频繁进行的编程作业的快速解决方法，以及为“如何解决问题”提供了清晰明了的约定。同时Django提供了CSRF的保护机制，防止例如恶意网站个指向你网站的链接。它的保护流程是第一次响应来自某个客户端的请求时，会在服务器端随机生成一个 token，把这个 token 放在 cookie 里。然后每次 POST 请求都会带上这个 token，这样就能避免被 CSRF 攻击。

1.4 Web前端实现

一个完整的网站包括前端和后台两个部分，简单来讲，后台是用来实现网站功能的，而前端则属于功能的表现，它直接影响用户的访问体验。前端页面是由一个个的html网页组成的，各网页之间通过超链接进行相互跳转，通过事件驱动JavaScript对用户的输入做出响应。

前端网页的开发主要使用HTML（超文本标记语言），CSS（层叠样式表）和JavaScript脚本语言。其中HTML用于编写静态网页，而CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。CSS具有精简代码，降低重构难度、提升网页访问速度等优势，能够对网页中的元素位置进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力。JavaScript是一种基于对象和事件驱动的相对安全的脚本语言,被广泛用于Web开发,常用来为网页添加各式各样的动态功能,为用户提供流畅美观的浏览效果。通常JavaScript脚本是通过嵌入在HTML中来实现自身功能的，该语言与操作系统无关，只依赖浏览器本身，如果使用的浏览器不支持JavaScript，那么嵌套在HTML中的JS代码则会被忽略。

在网站上使用样式表一共有三种方法：外部样式表、内部样式表和内联样式。

在本项目的Web网页开发中，用到了大量的HTML知识，我们直观看到的页面就是经过层叠样式表修饰之后的静态网页，而在页面中加载出来的文件则是通过脚本语言与后台交互调取出来的。本项目中使用的层叠样式表主要是外部样式表，通过使用外部样式表，只需要更改一个文件就能改变整个站点的外观，对于页面较多的网站非常有益。

1.5 文件操作的网页视图端实现

(1) 文件上传：

这部分内容是通过input file插件来执行文件上传的操作，大致流程是这样的：选择文件之后分块上传文件，因为文件是整个上传到主节点的，所以最后要根据上传的文件将整个文件分块存储到各个节点中，另外，我们用Django自带的数据库模型来存储上传文件的一些基本信息，如上传分块的时间，大小，存储的路径，哪一个节点等等信息。如果要上传带有中文的文件，需要修改配置文件，因为Django是默认语言是英语的，所以在setting.py文件中要把language的参数设置为“zh-hans”。

(2) 文件删除：

删除这部分功能也是一样的，因为在主节点上有一个主文件并且各个节点上有分块存储的文件及其备份，所以在做这部分的内容中需要将所有的文件都删除，另外，在上部分中提到我们在上传文件的过程中还在数据库中存储了上传文件的一些基本信息。所以在执行删除操作的时候还需要将数据库中该文件的相关信息给删除掉，不然再次上传同名文件的是会报错的。为了防止出现误删的情况，在点击删除的时候我们需要将提示用户是否确定删除该文件。另外，在获取文件名点的时候如果获取的文件名带有中文，那么读出来的字符串是会乱码的，需要将文件名以“GBK”的方式解码才行。

(3) 文件更新:

这部分内容因为要考虑到文件的一致性问题，所以他可以简单的分为两个部分：文件删除和文件上传。但是我们在执行这部分操作的时候需要了解需要更新的文件是什么和获取将要更新的文件，所以将获取的文件名参数放到需要提交的表单的参数中，由于这部分内容是动态更新的，加上文件更新和文件删除两个部分的功能是放在一个表单里实现的，跳转的路径也是不一样的，这部分功能需要在客户端用js来提交表单。

(4) 文件下载：

文件下载和文件上传的功能是类似的，因为上传的时候是根据10000B的块大小分块上传的，所以自下载的时候也是分块下载的，另外，就和文件删除功能一样，我们在获取文件名的时候如果是带有中文的话获取的字符串是会乱码的，所以需要将字符串以“GBK”的方式解码。

**二. 文件备份**

2.1 备份策略

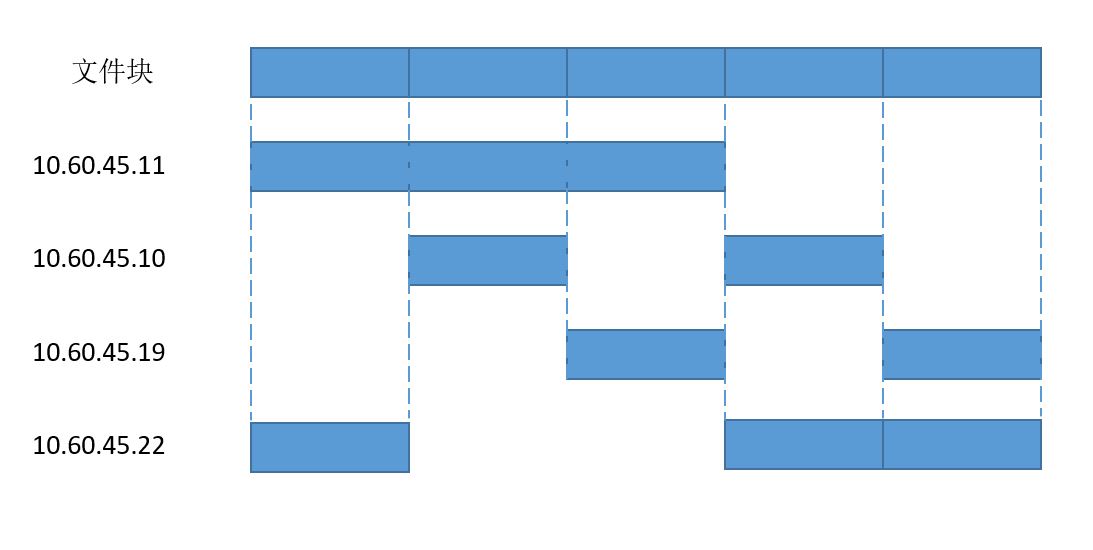
该系统采取文件分块存储策略。当文件上传至集群中的某一个节点后，文件会被分成若干个大小为1Mb的文件块，按分块顺序从0开始编号。分块完毕后，根据位图(Bitmap)索引将文件块传送至相应的节点。不属于当前节点的文件块在传送完毕后会被删除，每个节点只保留自身所对应的文件块。

备份完毕后，集群中所有的文件块可组成2份完整的文件，且任意一个节点宕机都不会影响文件的完整性。

2.2 位图索引

在设计文件块到节点的索引关系时，考虑到分布式系统存在大量的文件块查找工作，我们选用了查询效率高的位图索引方式。

文件块被分为五个区域，每个节点都负责存储和维护相应区域内的文件块。各节点对应的文件块范围如下图所示。



11节点存储文件块0,1,2；10节点存储2,4；19节点存储3,5；22节点存储1,4,5。每个文件块都在两个节点上存有备份，查询文件时会按顺序取第一个查询到的节点。若节点无响应，则顺序读取下一个节点。

因为文件分块存储并备份，所以当4个节点中任意一个节点宕机时，

**三. 一致性策略**

3.1 时间戳

每个文件的信息由数据库中的一张表DataFile来维护，表中记录了文件id，文件名，文件块个数以及时间戳。文件的状态以时间戳为准。每次进行文件更新操作时，都将当前时间记录下来，更新数据库表中的时间戳字段。

3.2 更新策略

在集群上，只维护文件的一个时间戳版本。

每当更新发生时，先比较当前时间与数据库中的时间戳。当时间戳落后于当前时间时，发生更新操作的节点通知其他节点删除原有的备份，然后存入最新的文件备份，同时将时间戳替换为当前时间。

每次文件操作通过判断时间戳来保证文件始终处于最新的版本，且多个节点上的文件版本保持一致。

**四. 分布式计算**

4.1 Spark 简介

Spark 是专为大规模数据处理而设计的快速通用的计算引擎，启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。它的性能特点主要体现以下四个方面：

* 速度快：内存计算下，Spark 比 Hadoop 快100倍；
* 易用性：Spark提供了80多个高级运算符；
* 通用性：Spark 提供了大量的库，包括SQL、DataFrames、MLlib、GraphX、Spark Streaming。开发者可以在同一个应用程序中无缝组合使用这些库；
* 支持多种资源管理器：Spark 支持 Hadoop YARN，Apache Mesos，及其自带的独立集群管理器。

Spark应用：

* 使用SparkConf()来创建SparkConf对象，在SparkConf对象上设置的任何参数都有高于系统属性的优先级。对于本项目，搭建了4个节点，一个主节点，3个子节点，设置主节点地址为主连接地址。
* 设置Spark主功能入口SparkContext，它表示连接到一个Spark集群并且能够用这个集群来创建RDD和广播式变量。
* RDD（弹性分布式数据集）在Spark中的基础抽象，表示一个不可变的、分区的元素集合，可以并行操作。

4.2 PySpark编程实现

Spark提供一个pySpark的编程接口，可以并行处理计算文件数据。根据本项目要求需要计算三个问题。

问题一：计算出用户的每日平均通话次数，并将结果以<主叫号码, 每日平均通话次数>的格式保存成 txt 或 excel 文件。

1. 读入数据并使用rdd.map()处理为分布式列表；
2. 使用rdd.map()函数记录所有数据中每个用户1次通话，由于可能每个用户可能通了多次电话，所以使用rdd.reduceByKey()函数记录所有数据中每个不同用户通话次数times；
3. 取出每个用户以及其对应的通话日期，用集合性质进行日期去重处理，然后使用rdd.combineByKey()函数记录每个不同用户以及其对应不同通话日期列表dates。
4. 使用rdd.fullOuterJoin()函数将数据格式变为[‘user’,(times,dates)],计算用户的每日平均通话次数：evert\_time=times/dates
5. 使用rdd.collect()函数将RDD所有元素返回一个列表list;
6. 将列表中的每个元素读入到txt文件中，结果以<主叫号码, 每日平均通话次数>的格式保存。

问题二：计算出不同通话类型（市话1、长途2、国际3）下各个运营商（电信1，移动2，联通3）的占比。

1. 读入数据并使用rdd.map()处理为分布式列表；
2. 使用rdd.map()函数获取通话类型数据与被叫用户；
3. 使用rdd.combineByKey()函数获取在不用通话类型下的各个运营商记录，数据结构为[‘通话类型’，[各个运营商记录]]；
4. 使用rdd.collect()函数将三种通话类型返回为3个List，通话类型市话1下的各个运营商列表记为List1，通话类型长途2下的各个运营商列表记为List2，通话类型国际3下的各个运营商列表记为List3;
5. 使用sc.parallelize()函数将List1处理为分布式列表，然后依次使用rdd.map()、rdd.reduceByKey()函数，处理为统计不同运营商次数，数据结构为[(‘电信1’，次数)]，[(‘移动2’，次数)]，[(‘联通3’，次数)]，使用rdd.collect()函数将rdd中的每个元素形成一个列表。List2,List3处理过程类似；
6. 计算不同通话类型下各个运营商的占比：由第5)中可知，计算在市话通话类型下，电信占比=电信次数/(电信次数+移动次数+联通次数)，移动占比=移动次数/(电信次数+移动次数+联通次数)，联通占比=联通次数/(电信次数+移动次数+联通次数)；在长途与国际通话类型下计算方法一致；
7. 将计算结果存入一个列表中，然后将列表中的每个元素读入到txt文件中。

问题三：计算出用户在各个时间段（时间段的划分如表 1 所示）通话时长所占比例，并将结果以<主叫号码, 时间段 1 占比, ..., 时间段 8 占比>的格式保存成 txt 或 excel 文件。

1. 读入数据并使用rdd.map()处理为分布式列表；
2. 定义函数1：将每个用户开始通话时间(00：00：00)转换为秒(s)，返回的数据结构为[‘用户’，开始通话时间]，然后用rdd.map()调用这个函数；
3. 定义函数2：计算每个用户在8个时间段的通话时长，返回的数据结构为[‘用户’，8个时间段的通话时长列表]，然后用rdd.map()调用这个函数；
4. 使用rdd.combineByKey()函数获取每个用户不同日期的8个时间段的通话时长列表，数据结构为[‘用户’，列表1，列表2，…，列表n]；
5. 定义函数3：将每个用户的8个时间段的通话时长相加，返回的数据结构为[‘用户’，整合8个时间段的通话时长列表]，然后用rdd.map()调用这个函数，使用rdd.collect()函数将rdd中的每个元素形成一个列表；
6. 计算用户在8个时间段通话时长所占比例；
7. 将计算结果存入一个列表中，然后将列表中的每个元素读入到txt文件中，结果以<主叫号码, 时间段 1 占比, ..., 时间段 8 占比>的格式保存成excel 文件。

五. 数据可视化

5.1 Echarts简介

ECharts，一个纯Javascript的图表库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），底层依赖轻量级的 Canvas 类库 ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。ECharts 提供了常规的折线图，柱状图，散点图，饼图，K线图，用于统计的盒形图，用于地理数据可视化的地图，热力图，线图，用于关系数据可视化的关系图，treemap，多维数据可视化的平行坐标。本次用到折线图，饼图，还有堆叠柱状图。

5.2 计算结果可视化

下载echarts.js文件，并与之后产生的.html文件置于同一文件夹，即简单配置好Echarts环境。如果后续还有其他使用要求，还有一些额外的包可以下载。

(1) Echarts图表的总体框架。

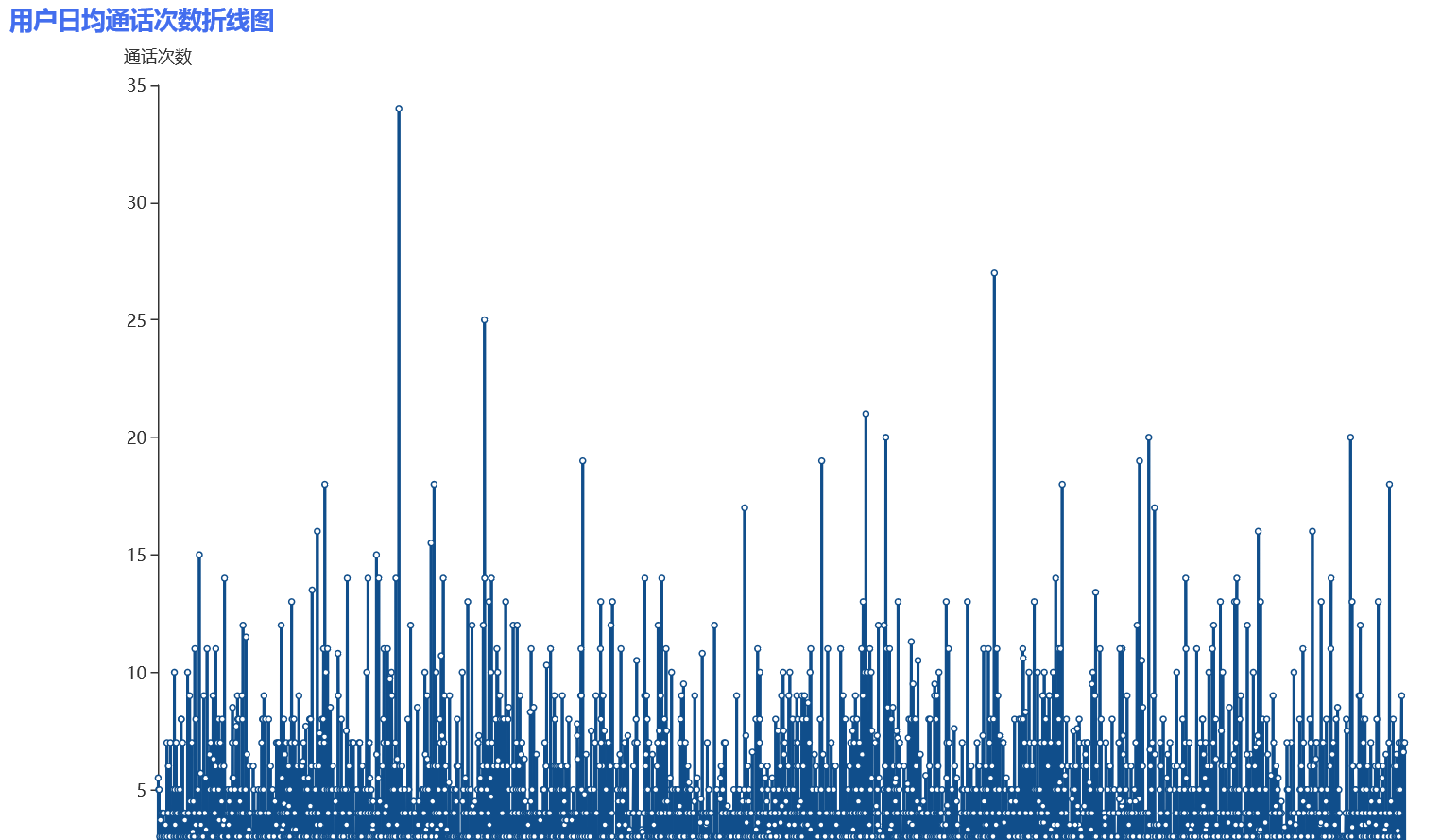
* 配置路径；
* 为Echarts准备一个具备大小（宽高）的容器；
* 指定图表的配置项和数据。加载不同模块，不同模块对应不同类型，如柱状图对应bar模块，折线图对应line模块，饼状图对应pie模块，按需加载；
* 函数调用；
* 使用指定的配置项和数据显示图表。

(2) 数据预处理：

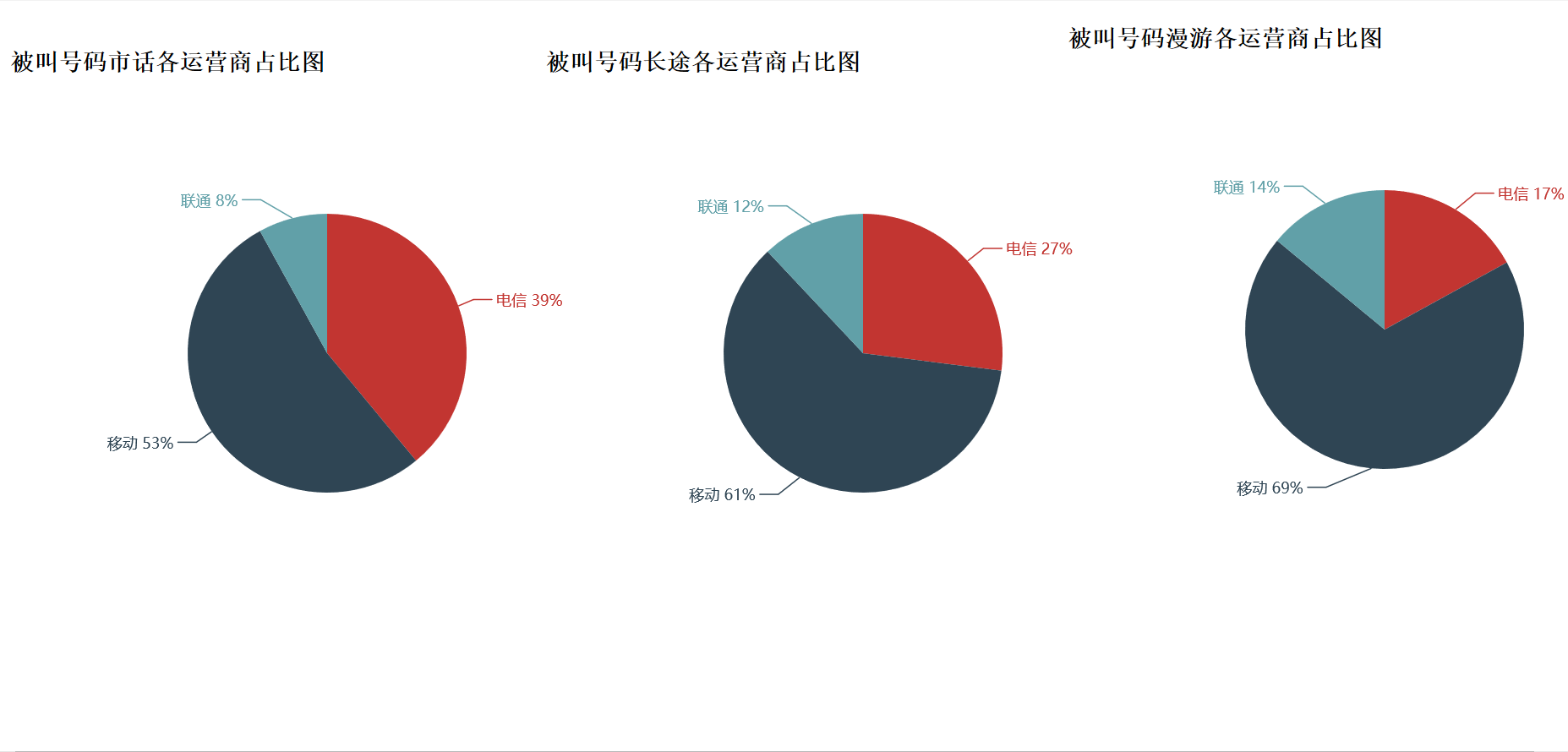
对于第一问和第三问，随机抽取五千条数据表示，因为数据量太大表现效果不太理想，并且excel的数据形式应转换为(‘用户名1’，‘用户名2’，…),对应的通话次数和通话占比的格式应转换为；(数字，数字，…)；

对于第二问，计算市话、长途、漫游条件下被叫号码的占比已有队友提供，故直接使用；

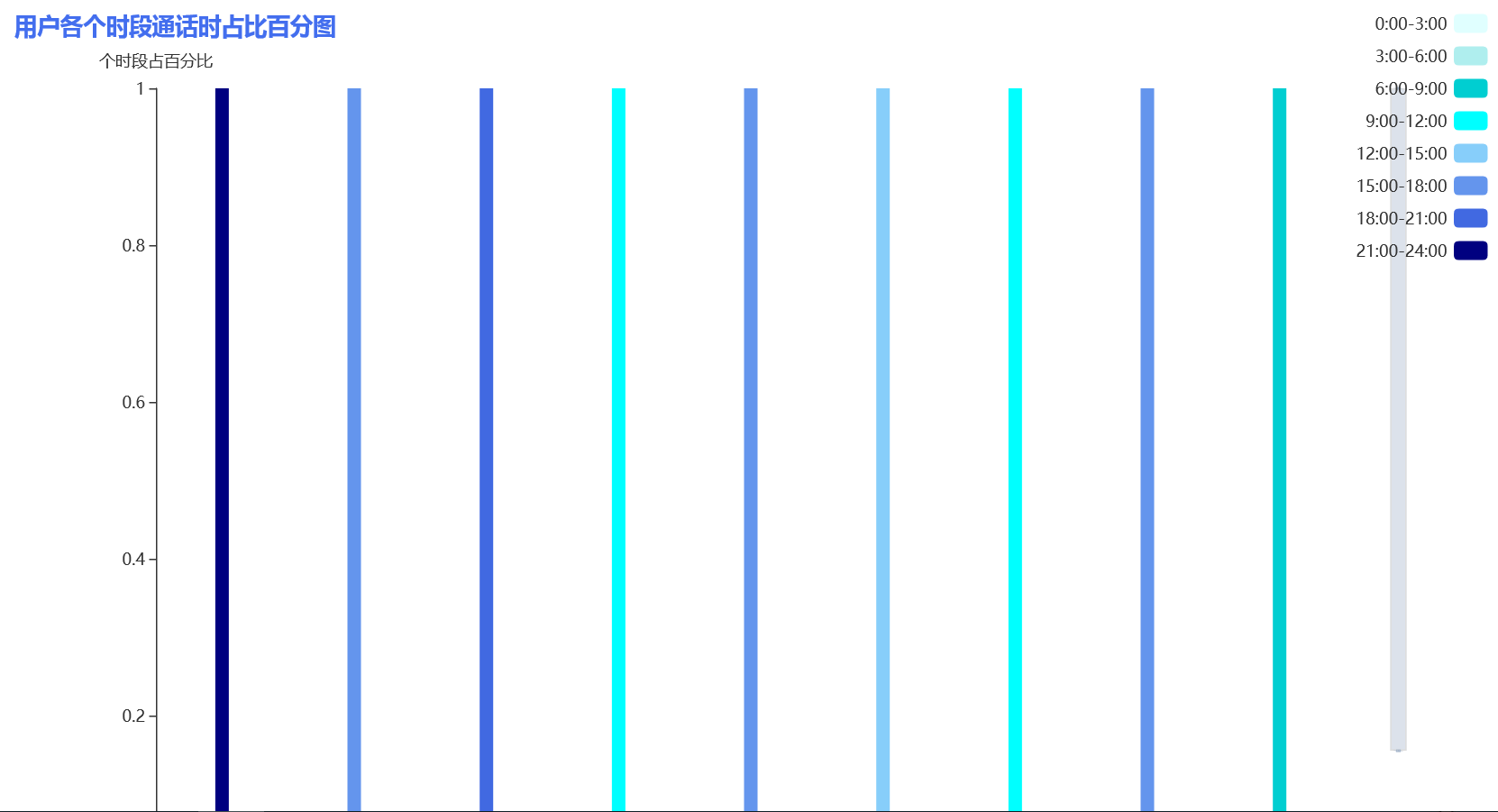
(3) 图表展示：



问题1：用户日均通话次数



问题2：运营商占比饼状图



问题3：通话时段占比图

第一张和第三张图加入Datazoom模块，可以使滑块在X轴水平滑动，Y轴自适应高度，方便展示更多的数据；

第三张图为了更加直观地表示各个时段的占比，从夜晚到白天，时间段所代表的颜色逐渐变浅，故可以从颜色深浅感受通话时段集中在什么时段，比较可惜的是很多用户一天只有一通电话，故最后的效果不是很明显；

同样为了显示更多的数据，在定义X轴的axislabel时，加入interval：0， rotate：-40。即可以将横轴显示，并且以-40度角倾斜。

六． 项目分工

1.李倩： Web前端页面设计与实现

2.朱西宁：数据可视化设计与Echarts编写

3.于陈峰：Django后端架构设计与视图逻辑编写

4.黎龙奇：Spark分布式计算编程

5.刘雅骏：集群搭建与文件分布式操作策略实现

6.项目答辩PPT与文档由所有组员共同完成