

**创新实验课程报告**

专 业 计算机科学与技术

学　　 号 1150310609

小 组 成 员 黄道龙（1150310613）

王陈阳（1150310609）

# 课题简介

## 1.1课题名称

分布式大规模网站访问日志管理

## 1.2课题任务

（1）假设一个网站有多个服务器，用户对网站的访问会随机分配到任意一台服务器。

（2）访问日志的格式 为 时间+用户IP的格式

（3）使用学过的数据结构和Maprduce架构，实现一个访问日志统计系统。

（4）实现日志汇聚，统计top k功能

（5）对极短时间内多次访问的用户，添加到临时黑名单结构中。

（6）自己构造足够量级的测试数据进行测试

# 运行环境

本次实验在WIndows 10系统下，利用Python 3.7语言进行开发测验。

# 数据生成

3.1数据存储格式

本实验的数据是模拟多个服务器被访问时产生的日志数据，所以生成的数据由多个文本文件保存，不同文本文件代表不同的服务器，并且每个文本文件的访问日志存储格式如下：

访问IP Access\_time

访问IP Access\_time

访问IP Access\_time

....................

其中访问IP的格式为标准的IPv4地址，Access\_time的格式是年-月-日-时-分-秒，即访问日志的时间的细粒度为秒。

3.2数据生成主要方法

3.2.1访问IP生成

通过随机生成在特定域的随机数，将其拼接为标准的IPv4地址作为本条日志的访问IP地址。

3.2.2访问时间的生成

先输入日志记录的开始时间及其日志记录范围，利用Unix存在的时间戳概念，将开始时间转换为时间戳。再随机生成日志记录范围内对应秒数的整数值，将两数相加，得到一个新的时间戳，该时间戳即为一个随机的访问时间，在将其转换为对应的访问时间的日志格式作为该条日志的访问时间。

3.3数据生成流程图

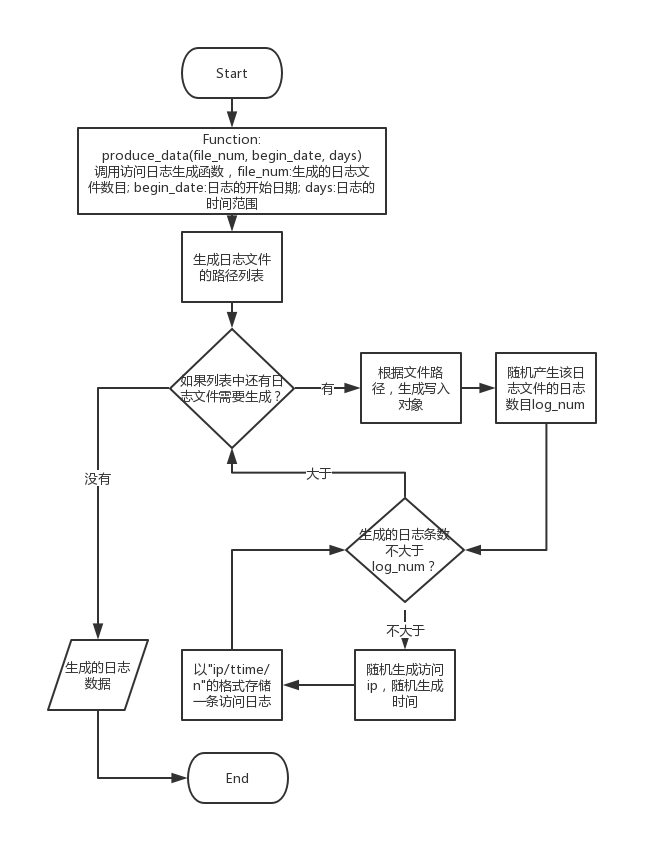


图1.数据生成流程图

3.4 数据的说明

本次实验中，生成的数据一共有100个文本数据文件，代表100个访问服务器的日志数据，日志数据一共有

# TopK

4.1主要方法

利用map-reduce的编程框架，对数据进行了两次map-reduce操作，将高频访问的IP按照访问次数排序输出。

4.2第一次map-reduce

4.2.1 map过程

读取生成的日志文本数据，将每一条日志文本数据：访问IP Access\_time以key:IP value：1的数据对存入程序中

4.2.2 reduce过程

将程序中存在的每一个key:IP相同的合并，其value为相同的IP的次数，即输出结果为：IP 访问次数

4.3 第二次map-reduce

4.3.1 map过程

将第一次map -reduce的结果以形式：key：访问次数 value：IP的数据对存储在程序中

4.3.2 reduce过程

利用map-reduce框架的自动排序，将访问次数高的数据对排序到前面，然后将其结果输入到一个文本文件中，该文件的内容即为所求TopK的次序

4.3程序流程图

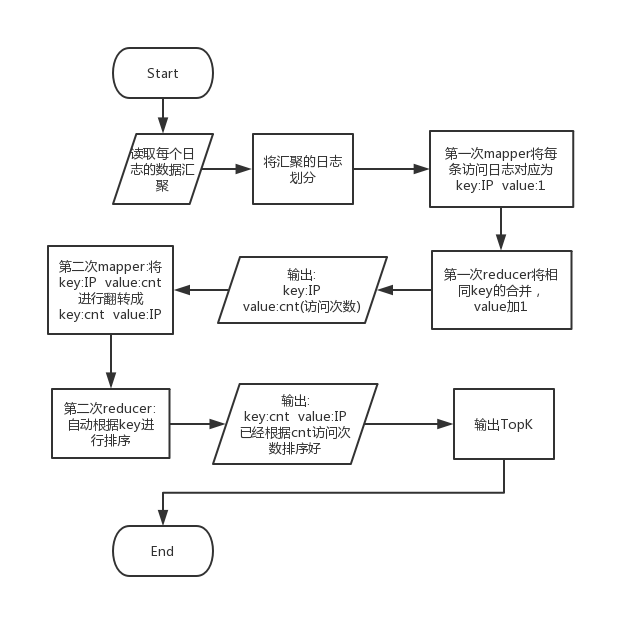


图2.TopK程序流程图

# Black\_list

## 5.1、功能描述

假设一个网站有多个服务器（实际随机生成了多个文本文件代替多个服务器上的日志文件），用户对网站的访问会随机分配到任意一台服务器，现在需要对极短时间内访问的用户临时黑名单结构中，即从多个不确定日志条数的日志文件中找出一定时间段内访问次数超过某一阈值的IP。

## 5.2、实现方式

①、使用mapreduce架构，对日志数据进行统计，设计相应的数据结构使得需要的数据能够进行排序，然后输出答案；

②、一定时间段内可以看作是一个滑动窗口，通过统计某一IP在该滑动窗口时间内的访问次数，即可确定该IP在该时间段内是否违反规定成为了黑名单用户。

## 5.3、第一次mapreduce

基本思想：聚合多个服务器上的日志数据，并按时间顺序对日志数据进行排序。

流程图：

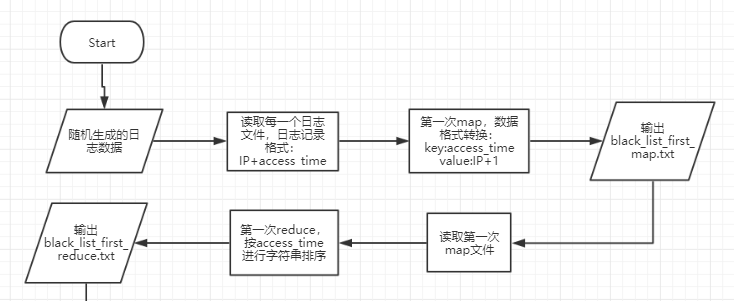


图3. 第一次map-reduce

（1）、map过程：将每一条日志记录映射成key—value结构，方便reduce过程

输入：每一条日志记录，格式：IP access\_time(制表符间隔)，

例：3.7.5.9 2019-05-01-00-07-44（制表符间隔）

输出：日志记录转换：key:access\_time，value:IP 1（制表符间隔）

例：2019-05-01-00-07-44 3.7.5.9 1（制表符间隔）

输出到black\_list\_first\_map.txt文件中。

（2）、reduce过程：读取black\_list\_first\_map.txt文件中的日志记录，以access\_time为键进行字符串排序，得到排序的日志记录，输出到black\_list\_first\_reduce.txt文件中。

## 5.4、第二次mapreduce

基本思想：map过程将每一条记录分解为一个热点时间记录（IP,ts,te,count）ts:访问开始时间，te:访问结束时间，count：访问次数，reduce过程按照事先约定的滑动窗口大小和访问次数阈值对日志记录中相同IP的记录进行统计和归并，找出满足黑名单条件的IP。

流程图：

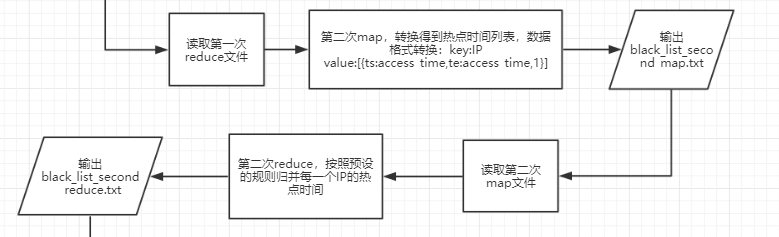


图4.第二次map-reduce

（1）、map过程：读取black\_list\_first\_reduce.txt文件，对每一条记录进行格式转换，映射成如下的key—value结构。

输入：第一次reduce后的日志记录（按访问时间排序）日志格式：access\_time IP 1

如：2019-05-01-00-07-44 3.7.5.9 1

输出：key:IP value: ts te count

例：3.7.5.9 2019-05-01-00-07-44 2019-05-01-00-07-44 1

分别对应开始时间（ts）、结束时间（te）、访问次数（count），输出到black\_list\_second\_map.txt文件中。

（2）、reduce过程：读取black\_list\_second\_map.txt文件，按照预设规则归并每一个IP的热点时间列表，并将满足预设规则的时间段及对应的访问次数添加到相应的结构中。

输入：第二次map后的日志记录，日志格式：IP ts te count

中间处理过程：

流程图：

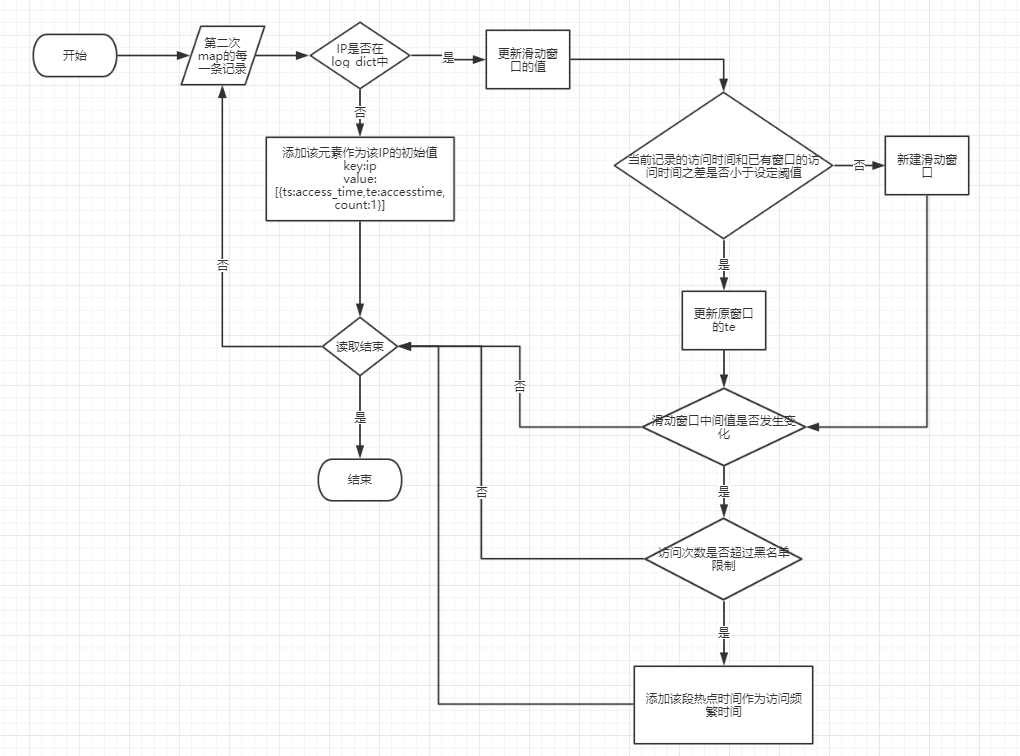


图5 第二次reduce

①、数据结构说明：log\_dict:存储IP及其热点时间段，key:IP，value:热点时间列表，热点时间列表中每一个元素是一个字典，字典中有三个键：ts，te，count（开始时间，结束时间，访问次数）

②、算法流程：

a、读取一条第二次map后的日志记录(IP ts te count)

a.a:判断IP是否在log\_dict中，若否，添加该IP及其对应键值后执行a，若是继续往下执行；

a.b:利用当前的日志记录与log\_dict[IP]进行比较，即调用更新滑动窗口的函数对log\_dict[IP]进行更新。更新规则：检查当前日志记录的开始时间和已有滑动窗口的中间时间节点的时间差是否大于规定窗口时间的一半，若是，则以当前日志的时间节点为新滑动窗口的中间时间节点，同时复制之前窗口内满足规定时间域的日志记录；若否，则填充当前的滑动窗口。

a.c:判断滑动窗口中间时间节点是否发生变化，若否，说明当前日志记录没有改变滑动窗口，则执行a；若是，则需要进行相应的条件检查，继续执行。

a.d:判断当前滑动窗口内的访问次数是否大于预设的阈值，若否，说明未达到触发条件，执行a；若是，则向log\_dict[IP]内添加当前滑动窗口的时间起始点作为热点时间域，当前滑动窗口内的访问次数作为热点时间域的访问次数。

b、读取日志记录结束

将log\_dict内的内容写到black\_list\_second\_reduce.txt文件中。

## 5.5、第三次mapreduce

基本思想：分解第二次reduce生成的每个IP热点时间记录，并进行相应的排序

流程图：

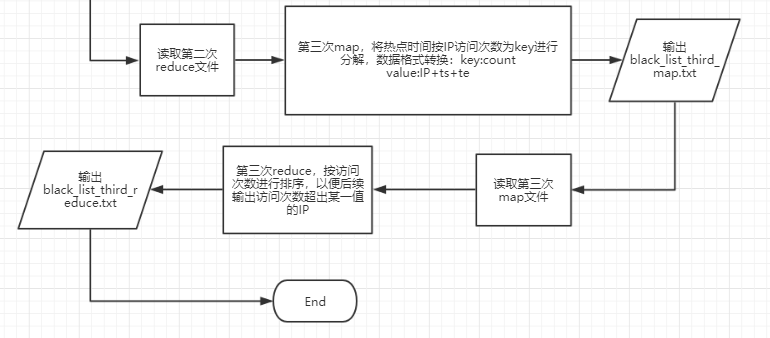


图6 第三次map-reduce

（1）、map过程：读取black\_list\_second\_reduce.txt文件，按照每个IP的每段热点时间的访问次数为key进行分解。

输入：IP [{ts:ts1,te:te1,count:count1},{ts:ts2,te:te2,count:count2},…]

输出：count1 IP ts1 te1

输出到black\_list\_third\_map.txt文件中。

1. 、reduce过程：读取black\_list\_third\_map.txt文件,然后以count次数进行降序排序，得到相应的日志排序记录，并输出到读取black\_list\_third\_reduce.txt文件中。读取black\_list\_third\_reduce.txt文件即可获取黑名单用户。

# 程序性能测试

6.1 TopK程序

6.1.1 程序运行结果截图

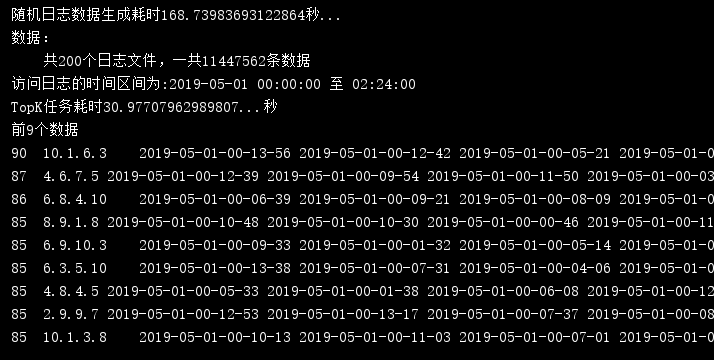


图7 TopK 运行结果截图

6.1.2 程序结果分析

在本次实验中，把IP的每个域值的随机生成区间限定在10以内。共生成了200个日志文件，一共11447562条日志数据。TopK任务的执行时间为30.98秒。

6.2 Black\_List程序

6.2.1 程序运行结果截图

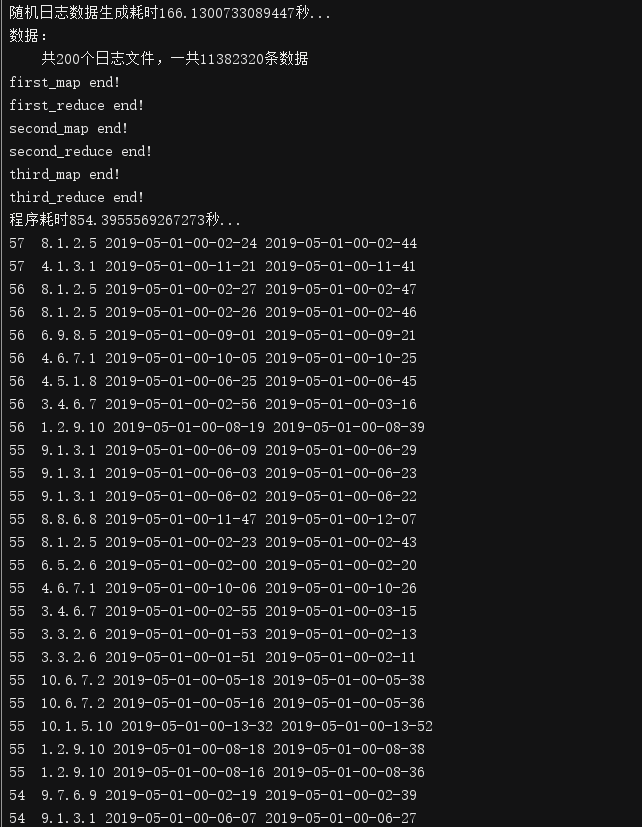


图8 Black\_List 运行结果截图

6.2.2 程序结果分析

在本次实验中，把IP的每个域值的随机生成区间限定在10以内，时间范围14.40分钟之内的访问日志，共生成了200个日志文件，一共11447562条日志数据。Black\_List任务的参数是20秒以内访问次数超过15次，该次任务的执行时间为854.40秒。