# 基于Hadoop的网站KPI指标统计

## 项目背景

很多场景下我们都需要统计一些KPI指标，如PV，UV，用于运营或市场分析，例如电商。

在互联网化的大背景下， 用户访问的终端越来越多，尤其在移动互联网高度发达的今天，用户终端数动则过亿，服务器端收集到的数据量也动则以TB、PB计。对于上述指标的分析，传统的分析手段难以胜任海量数据的场景。

本项目使用Hadoop的Map-reduce框架对上述海量数据分析的场景进行实践。

## 项目简述

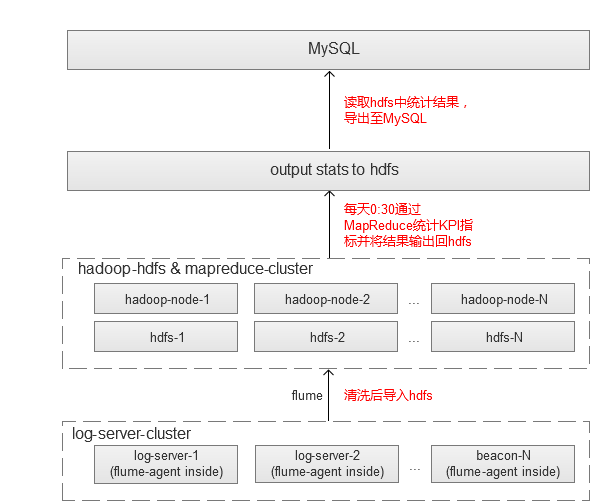
本项目通过解析前端Web服务器采集到的用户访问日志（日志中包含用户IP、请求URL、响应状态、来源Referer、UA等信息），完成多个维度的KPI指标统计，统计的指标及维度包括：

* PV(PageView): 页面访问量每日统计
* IP: 每日页面独立IP统计
* Time: 用户时间维度（每小时）PV的统计
* Source: 用户来源域名的每日统计
* Browser: 页面访问量按浏览器类型每日统计
* DeviceType: 页面访问量按设备类型每日统计

## 项目成员及分工

|  |  |  |
| --- | --- | --- |
| **学号** | **姓名** | **角色及分工** |
| GS1421877 | 廖雄杰 | 组长，架构设计，MapReduce开发 |
| GS132A56B | 林源 | 数据库设计，开发入库模块 |
| GS1421803 | 桑宏伟 | MapReduce开发 |
| GS1421833 | 司中原 | flume搭建，flume模块测试 |
| GS1421834 | 卞雪达 | MapReduce开发 |
| GS1421891 | 鞠光辉 | MapReduce开发 |
| GS1421892 | 张世杰 | mysql环境搭建，数据库模块测试 |
| GS1421912 | 王彬 | 框架搭建、设计、日志解析，清洗模块 |

## 系统架构



系统架构图

1. 用户上传的日志通过flume导入hdfs集群，导入的过程中同时完成数据清洗的动作，将爬虫或机器人（如Googlebot,msnbot,bingbot,360Spider等）从访问日志中过滤掉，以提高后续MR的效率，同时也尽可能排除爬虫或机器人访问对最终访问量统计结果的干扰。
2. 每天00:30运行统计Job扫描前一天导入的全部访问日志((hdfs://$HDFS\_PATH/data/input/kpidata/$DATE))，通过Map-Reduce将结果统计汇总，并将各维度的统计结果写回hdfs目录(hdfs://$HDFS\_PATH/data/output/kpidata/$DATE)。
3. Job全部执行完成后，再运行结果导出程序JobResultsExporter将统计结果导出至MySQL数据库中，供前端应用系统查询统计结果。

## 环境搭建

1. Hadoop集群搭建及部署

测试环境使用1台master + 5台slave

Hadoop使用官方最新版本:2.7.2

1. flume 环境搭建及配置

flume使用最新的1.6.0版本

**安装步骤**

执行命令解压安装包：

* [hadoop@cassdb ~]$ cd /home/hadoop/app
* [hadoop@cassdb app]$ tar -xzvf apache-flume-1.6.0-bin.tar.gz
* 使用文本编辑器对flume-hadoop.conf文件进行编辑，修改配置文件，sink端使用hdfs

主要配置：

agent1.sinks.hdfs\_sink.type = hdfs

agent1.sinks.hdfs\_sink.hdfs.path = hdfs://192.168.1.19:9000/data/input/kpidata/%y%m%d

agent1.sinks.hdfs\_sink.hdfs.filePrefix = test

agent1.sinks.hdfs\_sink.hdfs.rollSize = 65000000

agent1.sinks.hdfs\_sink.hdfs.rollInterval = 0

agent1.sinks.hdfs\_sink.hdfs.rollCount = 0

agent1.sinks.hdfs\_sink.hdfs.fileType = DataStream

agent1.sinks.hdfs-sinks.writeFormat = Text

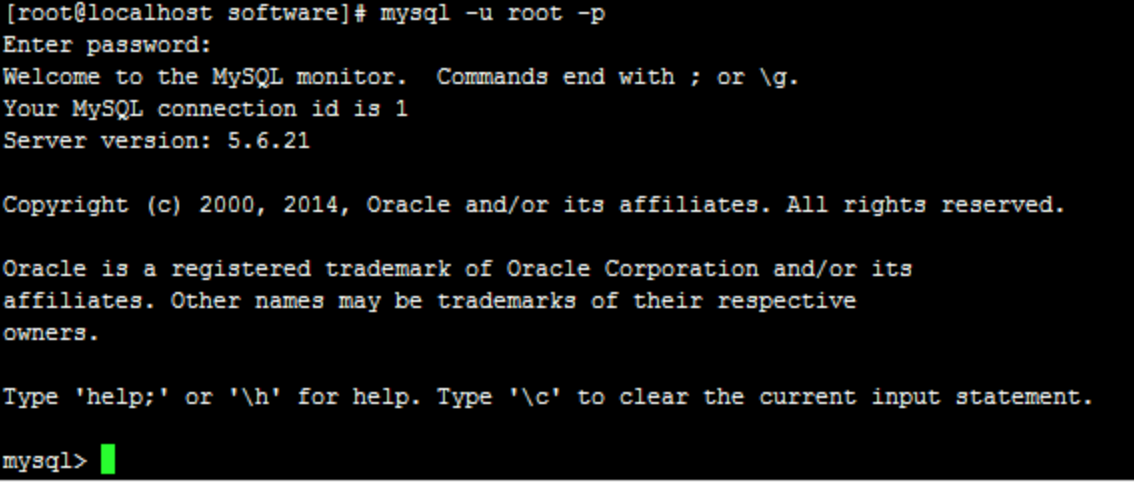
1. Mysql 环境搭建

**安装步骤**

执行命令解压安装包：

* [hadoop@cassdb ~]$ cd /home/hadoop/app
* [hadoop@cassdb app]$ tar -xzvf mysql-5.7.11-linux-glibc2.5-x86\_64.tar.gz

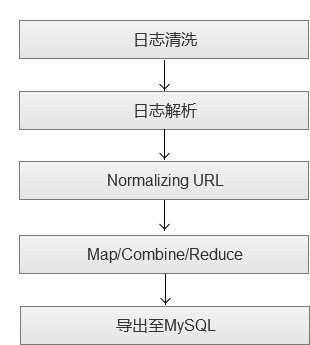
启动mysqld，并测试：

mysql -uroot -p -A

## 程序实现

统计全过程以键值对(Key-Value)为单元进行，以维度字段为Key(如日期、设备类型)，以统计指标为Value（如访问量）。

程序处理的主要流程如下：



以上模块各自功能如下：

**日志清洗：**

互联网上存在搜索引擎、网络爬虫或其它机器人，这些爬虫或机器人的访问数据也混杂在正常的访问请求中，例如下面的访问日志：

105.79.186.120 - - [01/Feb/2016:00:00:11 +0000] "GET /site/wiki/17 HTTP/1.1" 200 4333 "-" "Mozilla/5.0 (compatible; YoudaoBot/1.0; http://www.youdao.com/help/webmaster/spider/; )"

141.112.140.149 - - [01/Feb/2016:00:00:37 +0000] "HEAD / HTTP/1.1" 200 20 "-" "DNSPod-Monitor/1.0"

77.88.117.101 - - [01/Feb/2016:00:00:39 +0000] "GET /angularjs-route-template/ HTTP/1.1" 200 11933 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

在统计之前有必要先将这样的访问请求过滤掉，避免对统计结果造成干扰，另一方面，也能提高Map/Reduce阶段的效率。

清洗的过程通过关键字过滤，在flume的配置里增加一个按正则表达式过滤配置（Regex Filtering Interceptor）：

hadoop-practice.sources.r1.interceptors=intc1

hadoop-practice.sources.r1.interceptors.intc1.type=regex\_filter

hadoop-practice.sources.r1.interceptors.intc1.regex=".\*(robots.txt| Googlebot|bingbot|YandexBot|msnbot|YoudaoBot|DNSPod-Monitor|AhrefsBot|360Spider|YisouSpider|CompSpyBot|Sogou web spider|ia\_archiver).\*"

hadoop-practice.sources.r1.interceptors.intc1.excludeEvents=true

通过以上配置，flume将在导入hdfs前，将内容中包含robots.txt（爬虫协议文件访问请求）和其它包含机器人或爬虫关键字的请求过滤掉。

**日志解析：**

所有维度统计前的Map操作均需要解析日志文件，因此把日志解析单独抽象成公用的模块，供各维度统计模块调用。

解析成LogRecord对象，解析完成后提供以下字段：

time 请求时间

requestUrl 请求URL

httpStatus 响应状态

userAgent User-Agent头

browserName 浏览器名称

browserVersionName 浏览器版本

deviceType 设备类型：0 - PC 1 - Mobile

referer 请求来源Referer

**Normalizing URL:**

访问日志中很多URL和来源Referer都包含参数，甚至Restful的应用会将参数包含在URL路径中，进行页面及来源维度统计时需要将参数去掉或聚合，例如下面的URL：

http://blog.fens.me/wp-admin/post.php?post=2445&action=edit&message=10

/wp-content/uploads/2013/08/webstorm2.png

在进行URL及来源维度统计前，需要对URL进行Normalizing和适当的聚合，Normalizing的规则为：

1. 去除URL/Referer问号后的参数部分
2. 聚合路径中的疑似参数：将URL按路径分隔符(/)分段，若路径段为纯数字或十六进制数字或特殊分隔符（-\_\*+）。
3. 除了以上规则，也允许用户配置白名单规则，不经过上述Normalizing规则过滤。例如，有些电商的商品页，不同商品是通过参数来区分，例如京东的商品详情页：http://item.jd.com/1311928.html，用户可能希望统计某些重点商品或全部商品的PV等指标。

上面两个日志Normalize后变成：

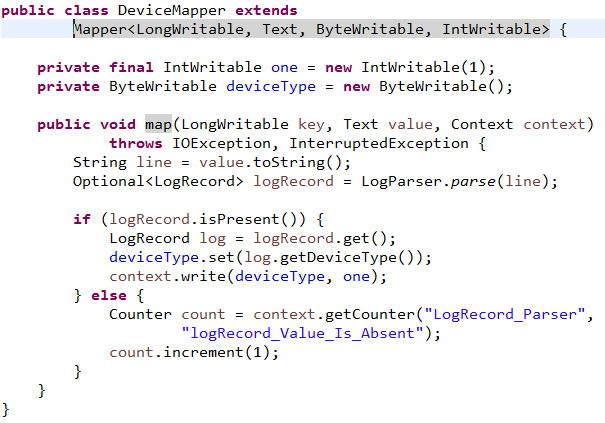
http://blog.fens.me/wp-admin/post.php

/wp-content/uploads/\*/\*/webstorm2.png

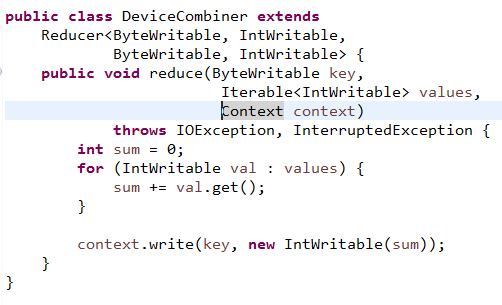
**MapReduce:**

在集群环境下完成各维度的指标统计，MapReduce的过程分为Map、Combine、Reduce三个阶段，中间加入Combine的过程主要是因为在海量日志的环境下，若Map后直接分发给各节点做Reduce操作，可能造成中间数据量太大，大量时间消耗在分发过程中的网络传输阶段，通过Combine在Map后立即进行本机预聚合，可大幅提高Reduce效率。

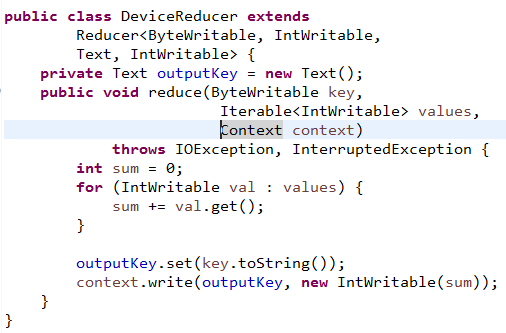
Mapper:



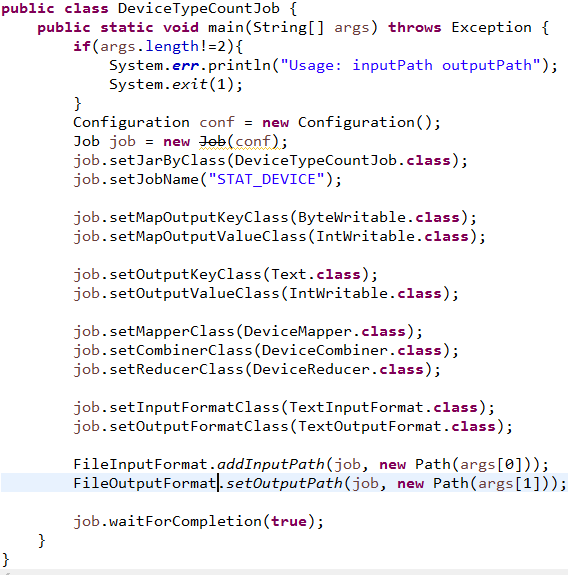
Combiner:



Reduce:



Job:

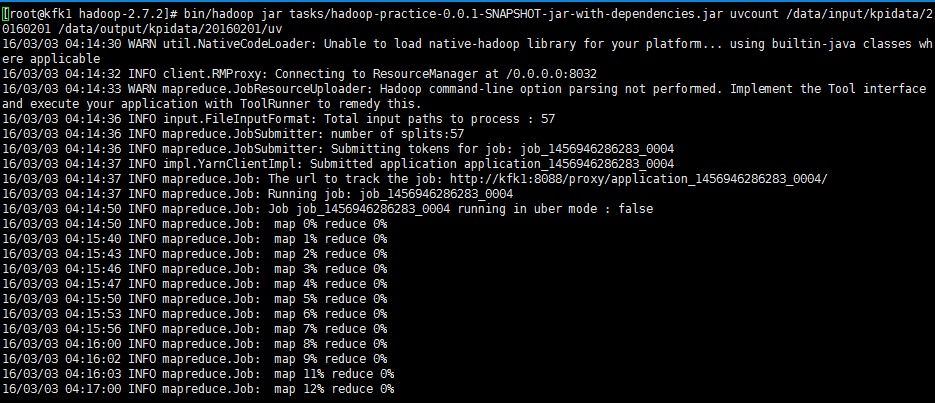


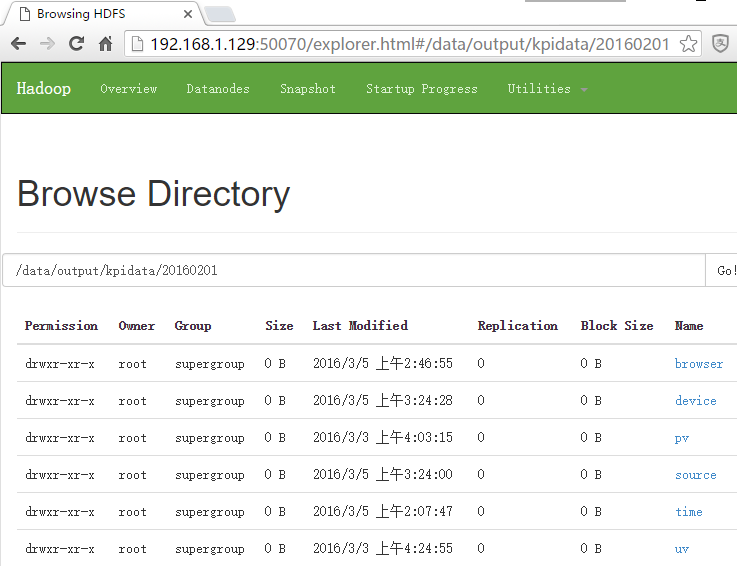
**导出至MySQL：**

MapReduce的统计结果存储于hdfs，JobResultsExporter负责将统计结果从hdfs导入至mysql。

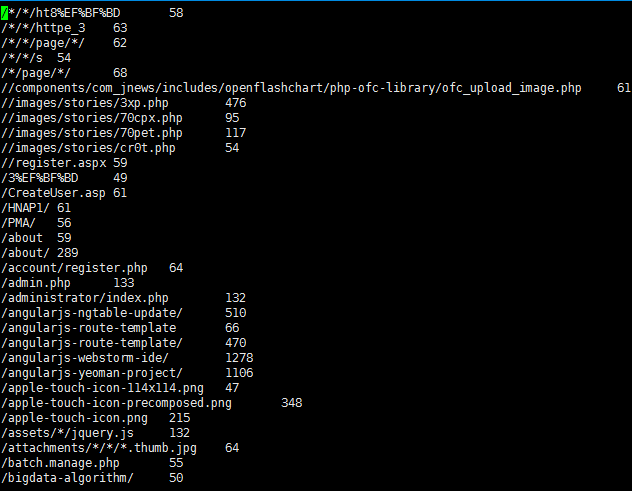


## MapReduce运行结果





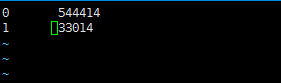
pv/part-r-00000:



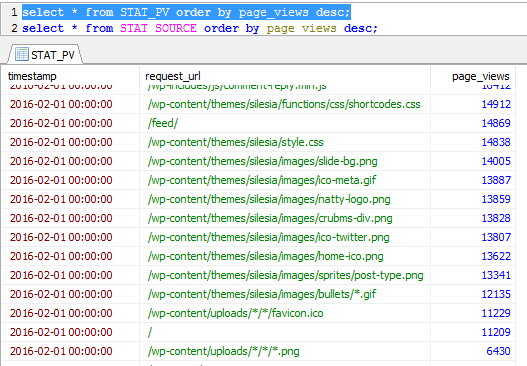
browser/part-r-00000:

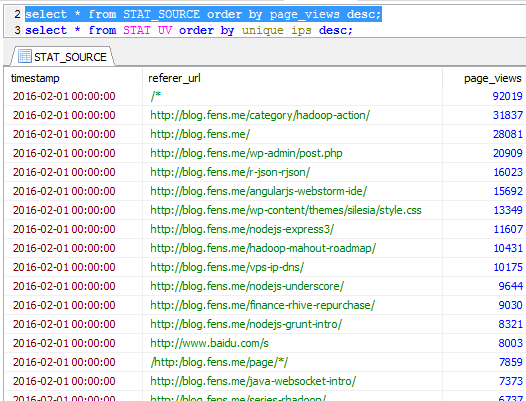


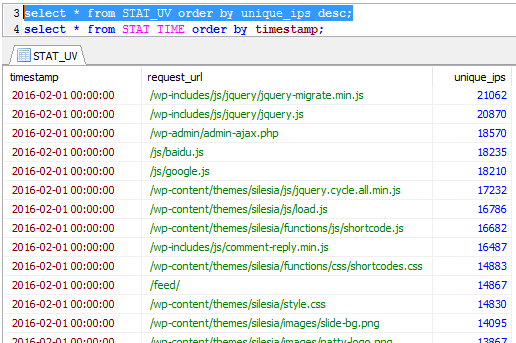
device/part-r-00000:

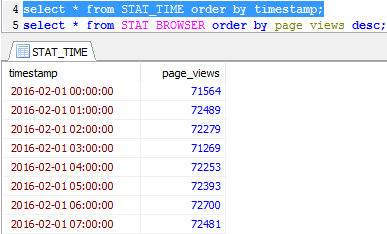


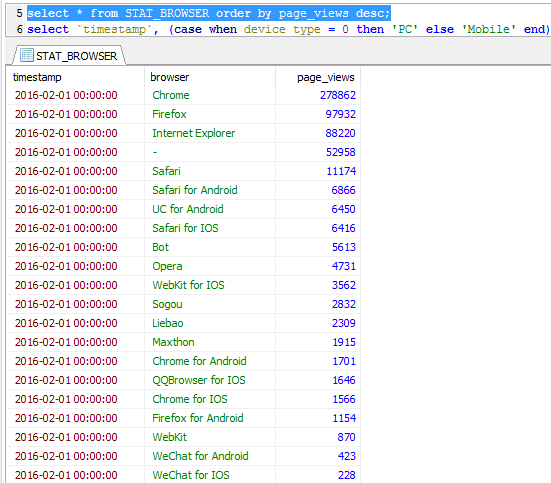
导入数据库之后的结果：

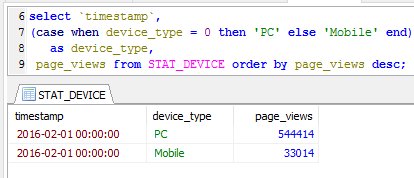






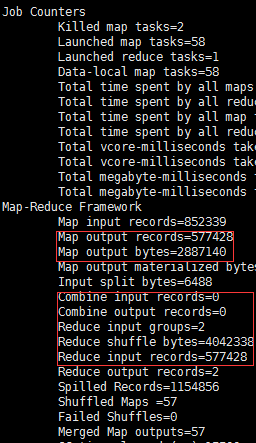
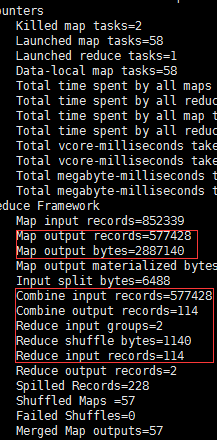






## 优化

1. **增加Combiner：**通过增加Combiner对Map结果进行本地预聚合，尽可能减少Reduce阶段网络传输的数据量

无Combiner 增加Combiner

1. **数据清洗：**避免爬虫和其它网络机器人访问请求对统计结果造成干扰，另一方面，也能提高Map/Reduce阶段的效率。
2. **UV统计优化：**对于海量访问的2C网站（例如双十一阿里的日访问用户可能过亿），常规的方法统计独立IP数（去重）可能会耗费大量内存资源，即使通过hadoop分布式处理，也需要大量节点支撑该指标的统计。通过集体讨论及查阅相关资料，我们选择HyperLogLog算法对独立IP数进行统计，该算法是可以以极少的内存（几十KB即可）样本集合的基数进行估算，误差可控制在1%以内，能满足绝大部分场景的需求。