日志收集模块说明

日志收集的目的最终是把用户对目标网站的访问日志汇聚到HDFS文件系统中的特定目录，以便提供给下一步的数据清洗模块进行处理。要完成这一工作，需要分为如下几个步骤去做：

* JS埋点
* 日志服务器搭建
* 日志切割
* 日志收集

下面对上述四个步骤一一进行详细说明：

1. **JS埋点**

我们需要收集某个第三方访站的访问情况，通用的做法是在第三方网站的相关页面嵌入一个JS脚本，这样当用户访问该页面时，页面上的JS脚本会在页面上动态加入一个<image/>标签，并且<image/>的src属性指向日志服务器下的一个透明图片的URL地址，在该URL参数上会附带第三方网站的一些信息（如被访问页面的URL、识别用户身份的cookie等）,这样通过分析日志服务器（一般是nginx或apache）的access日志文件就能获取到这些信息。这个JS埋点的脚本可以自己编写来实现，代码片断如下：

**function** ar\_main() {

**var** unit\_id = "9999"; **//第三方网站id**

**//日志服务器地址**

**var** dest\_path = "http://192.168.42.130/logtest/images/blank.gif?unit\_id="+unit\_id;

**var** expire\_time = 1800;

**var** i;

**//通过cookie获取UV信息**

**//返回导航到当前网页的超链接所在网页的URL**

**//获取当前访问页面的URL**

**//获取屏幕信息**

**//获取是否使用JAVA和flash**

**......**

**//拼接访问日志服务器的URL**

dest=dest\_path+"&url="+url+"&urlname="+urlname+"&title="+title+"&chset="+charset+"&scr="+screen+"&col="+color+"&lg="+language+"&je="+jvm\_enabled+"&ec="+cookie\_enabled+"&fv="+flash\_ver+"&cnv="+String(Math.random())+"&ref="+ref+"&uagent="+agent+"&stat\_ss="+uv\_id+"&stat\_uv="+stat\_uv;

dest+=ar\_join\_clientInfo();

**//在当前用户访问的页面上动态添加一个 <image/>标签，并设置其src为日志服务器下的一个透明图片的URL**

**var** dom = $("<img src=\""+dest+"\" border=\"0\" width=\"1\" height=\"1\" />");

document.body.appendChild( dom.get(0) );

}

**//执行脚本**

ar\_main();

------------------------------------------------------------------------------------------------------------------------------------------------------------------

此外这个js埋点的脚本我们也可以考虑用第三方开源的产品，这样这些开源产品会自动帮助我们出分析结果，这样的产品有：

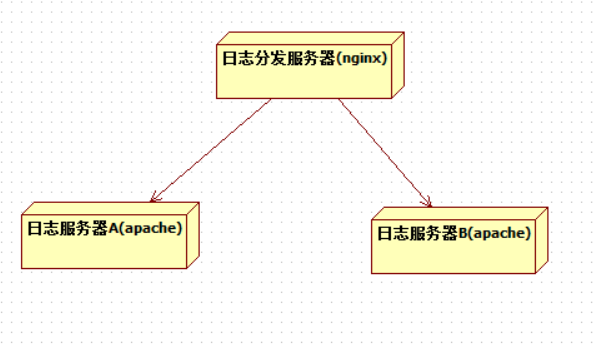
谷歌的Analytics ，其源码地址：<https://github.com/googleanalytics> ，

使用说明：<https://developers.google.com/analytics/devguides/collection/analyticsjs/>

腾讯云分析平台，http://mta.qq.com/

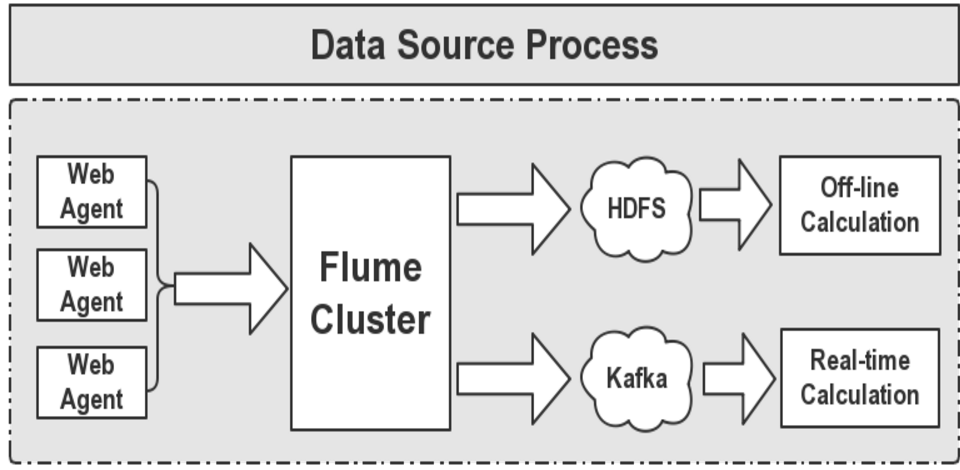
1. **日志服务器搭建**

在实际应用中，日志服务器应为多台，部署成负载均衡的方式。其中nginx作为分发器，用户的日志请求会发送到nginx之上，然后再通过其分发到一台日志服务器。日志服务器采用apache来搭建，这是考虑到apache的模块对日志的切割支持得比较好。



1. **日志收集**

Apache等Web服务器中的访问日志文件默认情况下的路径是/var/log/httpd/access\_log，可以通过/etc/httpd/conf/httpd.conf进行配置。默认情况下，apache会把用户的访问记录不断写入到access\_log文件中，这们就不方便flume对日志进行收集，所以根据实际业务需要，我们可以制定两种日志收集策略：实时收集和定时收集。



首先要先安装Flume ，下载地址：<http://www.apache.org/dyn/closer.lua/flume/1.6.0/apache-flume-1.6.0-bin.tar.gz>

下载后解压后即完成安装

* **实时收集**

需要实时监控日志文件access\_log文件的变化，在linux下可以使用tail命令来达到这个目的。

flume.conf配置：

tailAgent.sources = execSrc

tailAgent.channels = memoryChannel

tailAgent.sinks = loggerSink

tailAgent.sources.execSrc.type = exec

tailAgent.sources.execSrc.channels = memoryChannel

tailAgent.sources.execSrc.command = tail -F /var/log/httpd/access\_log

tailAgent.sinks.loggerSink.type = logger

tailAgent.sinks.loggerSink.channel = memoryChannel

tailAgent.channels.memoryChannel.type = memory

tailAgent.channels.memoryChannel.capacity = 100

启动tailAgent的命令：

{$FLUME\_HOME}/bin/flume-ng agent --conf ../conf/ -f ../conf/flume.conf -Dflume.root.logger=DEBUG,console -n tailAgent

这样每当/var/log/httpd/access\_log文件中出现追加内容时，flume都能实时捕捉到

* **定时收集**

需要按照定时的周期对日志进行切割，当前需求下也就是每小时需要产生一个日志文件。然后配置flume监控日志文件所在的目录。每当目录下出现新的日志文件，都会被flume的agent发送到HDFS的系统上。下面对这几个步骤依次进行分析：

* + 日志切割

apache服务器大约每响应一万次请求，access log文件会增长1M，这样这个access\_log文件就会逐渐变得很大。针对这个情况，Apache Server提供了一个Log Rotation功能，能够定期产生新的日志文件。在实际应用时Apache httpd能够通过管道的方式来调用Log Rotation进程。具体需在/etc/httpd/conf/httpd.conf文件中进行配置，

<IfModule log\_config\_module>

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined //定义日志格式和别名combined

LogFormat "%h %l %u %t \"%r\" %>s %b" common //定义日志格式和别名common

<IfModule logio\_module>

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio

</IfModule>

#CustomLog "logs/access\_log" common

CustomLog "logs/access\_log" combined //CustomLog是配置日志生效的指令

</IfModule>

上面是apache的默认配置，为达到切割效果，CustomLog指定的配置要改成下面这行：

**CustomLog "|/usr/sbin/rotatelogs -c -L /etc/httpd/logs/access\_log -p /etc/httpd/logs/log-executor.sh logs/access\_log\_%Y\_%m\_%d\_%H\_%M.log 3600 480" combined**

在上述配置中3600代表秒，表示每隔一小时生成一个日志文件。480表示相对于UTC的时差的分钟数。如果省略，则假定为"0"并使用UTC时间

-c表示表示如果没有请求时也会创建空的日志文件

-p 表示在产生新的日志文件时rotatelogs会调用我们指定的程序，并且会为这个程序隐含的传两个参数，第

一个参数是生成的新生成的日志文件的名字，第二个参数是上次生成的日志文件的名字，利用这一点，

我们可以写一个脚本来实现把前一个已完成的日志文件在此时copy到一个与flume采集相关的目录中，

以便flume进行采集，我们在此处插入的脚本代码如下：

newLog=$1

oldLog=$2

sourfile="$oldLog"

destfile="/user/root/logs/"

cp $sourfile $destfile

**注意事项**：chmod o+w /var/log/httpd/ 要把日志目录的可写权限打开，这个写入是由apache组的用户在执行。

* + flume收集配置

a1.channels **=** ch-1

a1.sources **=** src-1

a1.sources.src-1.type **=** spooldir

a1.sources.src-1.channels **=** ch-1

a1.sources.src-1.spoolDir **=** /var/log/apache/flumeSpool

a1.sources.src-1.fileHeader **=** true

1. 根据ip地址获取运营商及相关地域信息

可以通过淘宝IP地址库提供的API来实现 <http://ip.taobao.com/ipSearch.php>