hadoop的离线大数据分析平台项目

1. 企业项目业务设计：学习企业中常见的业务需求。
2. 数据采集
3. 数据清洗
4. 数据分析
5. 数据展示

企业项目业务设计

开发流程：

项目调研，需求分析，方案设计，编码实现单元测试，测试（功能测试，联合测试，用户测试，压力测试），部署上线，远行维护

企业大数据应用领域

1. 数据银行：以出售数据为业务核心的公司
2. 数据分析（数据分析平台）：一般来说我们都是针对业务自建数据分析，既能满足个性化需求，又能保证数据安全。
3. 搜索引擎：

   solr（https://blog.csdn.net/luo609630199/article/details/82494708）

   Lucene（https://blog.csdn.net/luo609630199/article/details/81839513）（https://blog.csdn.net/weixin_42633131/article/details/82873731）

   ELK：elasticsearch开源分布式搜索引擎+logstach收集数据+kibana数据web展示（https://blog.csdn.net/zhangxtn/article/details/51454881）（https://blog.csdn.net/longxibendi/article/details/35237543/）
4. 推荐系统：mahout,sparkMLlib
5. 精准营销：广告投放，金融投资
6. 数据预测：天气预测，路况预测，城市发展相关（人口，交通等）预测
7. 大数据征信（要有征信牌照）：芝麻信用，借钱以及免押金的凭借、腾讯征信
8. 人工智能：算法-python,机器学习-python，深度学习-python
9. 用户画像：为用户打标签，电商常用

企业大数据分析平台

1. 目的：数据分析，提供业务支持及高层决策。
2. 离线数据分析：对分析的数据结果时效性要求不高，比如网站运营指标分析。

   技术：主要是mr/hive/pig/impla/spark on yarn,目前企业中也逐渐采用spark进行快速离线数据分析

   硬件要求：硬件要求不高，拿时间换空间

3. 实时数据分析：对分析的数据结果时效性要求高，比如商品推荐。

   技术：sparkstreaming/storm

   硬件：要求高，空间换时间
4. 离线和实时进行组合：实时分析，立即反馈；离线分析校正实时分析。
5. 和商业数据分析平台对比

   自建：

   商业平台：
6. 数据来源

   服务器数据：Nginx,Apache,Linux服务器日志

   业务数据：一般是存储在RDBMS中元数据，比如用户，订单，商品

   用户行为数据：用户访问行为的数据

   爬虫数据：网络爬虫

   购买的外部数据：银行数据，广告数据
7. 数据采集：

   sqoop采集关系型数据库中数据

   flume流式数据采集框架
8. 数据存储：

   基于磁盘的分布式文件系统HDFS

   基于内存的分布式文件系统Tachyon
9. 数据清洗与数据分析：

   ETL：字段过滤，字段格式化，字段补全

   分析：根据业务规则，计算数据指标值。

   存储：RDBMS，nosql
10. 数据实时高效查询：使用solr等搭建快速搜索应用，使用hbase等nosql数据库实现实时查询
11. 数据应用：搜索引擎，推荐系统，精准营销，机器学习与人工智能

数据流量及集群规划

数据量：决定了集群的规模

- 字段个数
- 每天处理的记录数（大型网站：千万级别数据访问记录；中小型网站：百万级别的访问量。）
- 处理数据的总量（每条记录大小*每天的记录数*360*2年）
- 保存的时间，一般是保存2-3年
- 计算每台服务器存储数据量：

  ①常见的每台机器磁盘大小是16T（8槽x2T磁条）；

  ②一般的磁盘利用率不超过80%，也就是说每台机器存储12.8T

- 计算需要多少台机器进行存储（DataNode节点个数）

  ①中小型集群：20~50台机器；

  ②中型集群：50~100台机器；

  ③大型集群：100~台以上

集群规模

（1）DataNode/NodeManager：25台

（2）NameNode/ResourceManager:2台（active standby HA）

（3）Zookeeper:3台

（4）JournalNode:3台

（5）Hive/sqoop：1台

◆一条日志记录大小约为300~500bytes

◆以网站访客日均 1000万，平均每人访问5次，每次访问5个页面为例

◆一日的日志数据量为：300~500 * 1000 0000 * 5 * 5/ (1024 * 1024*1024) = 70~117GB

◆ 要求保存三年历史数据：365 * （70 ~ 117 GB） * 3 （hdfs备份数）* 3 = 225~ 384 TB

◆ 服务器一台 磁盘一般 16TB，存储时最大存储 80%，也就是一台服务器存储量为 12.8 TB

◆ 需要的datanode节点数量：17 ~ 30 台，加上 zookeeper 3台，NameNode HA 2台，JournalNode 3台

◆ 集群总共需要 21~ 37台

资源配置（每台机器）

- CPU：决定了应用程序的快慢

  物理棵数和核数，核数越多越好，大数据计算都是多进程，多线程的并发处理，所以计算快慢由CPU的总核数决定

  分配：一个核数远行1-2个线程任务，具体配置根据业务决定

  常见配置：16-32-64核数

- memory

  常见配置：32/64/128G

  yarn：32G-64G

  spark：64G-128G

  内存大小是CPU核数的2倍=CPU核数*2

内存分配

NN:16G;

RM:2G;

Hmaster:2G;

DN/JN/NM:1G;

HregionServer:16G;

Hive/sqoop/ZK:1G;

Yarn资源调度：

每个NodeManager任务默认的分配内存（在yarn-default.xml中）是：yarn.nodemanager.resource.memory-mb=8G，我们一般会设置为 16G~32G.

每个NodeManager任务默认分配的CUP核数（在yarn-default.xml 中）：yarn.nodemanager.resource.cpu-vcores=8，一般设置为内存大小

yarn默认分配是1核1G内存

- 磁盘：一般8块磁盘，每块1T~2T，ZK要求的磁盘性能高
- 网络：千兆网络~万兆网络，万兆交换机，最好不要跨集群。
- job规划-以分析前一天的数据为例：

  ①job每天40个左右

  mr每天30个左右

  hive每天10个左右

  ②job运行时间

  ETL：10分钟

  分析job: 30分钟

  ③job调度工具实现job并发运行

企业常规分析需求

需求分析一：

电商上线后，通过收集用户行为数据，进行多维度统计分析，掌握网站线上运营情况，供运营部门分析业务展开情况，以优化网站，调整广告投入，进行更好的促销，精准营销等活动

往往由公司运营部门提出

需求分析二：

电商网站核心关注点：

购买率，复购率，订单数量/金额/类别情况，成功支付订单数量/金额/类别情况，退款订单数量/金额/类别情况，访客/会员数量，访问转会员率（新访客和老访客的转会员率），广告推广效果，网站内容吸引度（跳出率）

重要概念：

用户，会员，PV（Page view）网页浏览数，UV（unique visitor）网站独立访客，会话与会话时长，DV（网站访问深度），跳出率，维度信息（时间维度，平台维度，浏览器维度，地域维度，kpi维度）

最终功能构成：

用户基本数据分析模块，用户分析，会员分析，会话分析，HOURLY分析，浏览器信息分析模块，地域信息分析模块，用户访问深度分析模块，外链数据分析模块，订单数据分析模块，事件分析模块

FLUME实时流式数据采集

flume的介绍及组成

Flume功能

Flume是一个分布式，可靠的，可用的，非常有效率的对大数据量的日志数据进行收集，聚集，移动信息的服务。Flume仅仅远行在linux环境下。

他是一个基于流式的数据的非常简单的，灵活的架构，它是一个健壮的，容错的。它用一个简单的扩展数据模型用于在线实时应用分析。

简单表现为：写个source，channel，sink，之后一条命令就能操作成功了

Flume,kafka实时进行数据收集，spark，storm实时去处理，impala实时去查询

Flume特点

（1）用于数据的手机、聚合、移动

（2）基于数据流，用于在线实时的数据分析，比如双十一

（3）flume只能运行在Unix环境中。如果数据在windows环境中，该如何采集？

（4）简单，只需要运行一个配置文件集合

①flume-ng只有一个角色agent，可以在配置文件中配置多个agent，在每一个agent定义source，channel，sink三大组件。

②采集的时候要确定采集源和目标

③source 负责收集数据，并把数据发送给channel；channel负责临时存储数据；sink负责从channel取数据，并把数据发送到目标

source监控某个文件，将数据拿到，封装在一个event当中，并put/commit到chennel当中，chennel是一个队列，队列的优点是先进先出，放好后尾部一个个event出来，sink主动去从chennel当中去拉数据，sink再把数据写到某个地方，比如HDFS上面去。

EVENTS

event是flume数据传输的基本单元

flume以事件的形式将数据从源头传送到最终的目的

event由可选的header和载有数据的一个byte array构成

载有的数据对flume是不透明的

Header是容纳了key-value字符串对的无序集合，key在集合内是唯一的

测试远行

https://blog.csdn.net/qq_43193797/article/details/86572149

（1）source

①exec：tail

https://blog.csdn.net/liuxiao723846/article/details/78133375

https://blog.csdn.net/wuxintdrh/article/details/63787798

业务场景：用于动态监控某个文件。

几种flume监控方式：https://www.jianshu.com/p/09493efe0fb8

②spooldir，动态监控某个目录 https://blog.csdn.net/wuxintdrh/article/details/79478710

③kafka https://www.cnblogs.com/cnmenglang/p/6550427.html

④http https://www.cnblogs.com/duaner92/p/10114350.html

⑤syslog https://blog.csdn.net/day_one_step/article/details/75434731

（2）channel

①memory https://blog.csdn.net/ty_laurel/article/details/53907926

②file https://blog.csdn.net/tian_qing_lei/article/details/77725762

http://www.cnblogs.com/yurunmiao/p/5603097.html>

③kafka https://blog.csdn.net/wangshuminjava/article/details/80551314

（3）sink https://www.cnblogs.com/swordfall/p/8157766.html

①HDFS

②Hbase

③Hive

Flume企业常用案例

1.从hive读取日志到flume日志，并打印出来

（1）source：exec

（2）channel：mem

（3）sink：log

（4）运行 bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/hive-mem-log.properties -Dflume.root.logger=INFO,console

（5）常见的channel：memory/file，memory的capacity/transanctionCapacity最好在1/10到1/100之间

2.channel换成file

（1）运行， bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/hive-file-log.properties -Dflume.root.logger=INFO,console

（2）定义 checkpointdir和 dataDirs

3.sink更换为hdfs

（1）控制文件大小，发现生成的文件不是我们设置的文件大小，比如我们设置10kb，那么生成的文件在11kb左右，因为保证event的完整性。所以我们设置文件大小为128M的时候，只能设置为120M的样子。

a1.sinks.k1.hdfs.rollInterval = 0

a1.sinks.k1.hdfs.rollSize = 10240

a1.sinks.k1.hdfs.rollCount = 0

（2）按天分离文件

配置a1.sinks.k1.hdfs.path = /flume/tailout/%y-%m-%d/%H%M/，这样的话导入Hive中的时候可以按天导入

（3）运行 bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/hive-mem-hdfs.properties -Dflume.root.logger=INFO,console

4.动态监控文件夹

（1）应用场景是每天生成的日志文件的名称不同，比如20181101.log

（2）动态过滤，只监控设置的过滤文件。ignorePattern,includePattern

a1.resoures.s1.ignorePattren=([^]*.tmp$)就是用正则表达式过滤掉以.tmp结尾的文件，应用场景是日志文件在某一个时刻生成.tmp临时文件，在某个时刻重命名为.log文件的场景。

（3）场景，在某个时刻在目录中生成了一个.log文件，比如20181101.log，然后不停地追加数据，直到某个时刻停止，比如20181102，有生成了另外一个.log，比如20181102.log文件.spooldir不能解决这个问题，因为不但要动态监控目录，还要动态监控文件。这个时候就会用到taildir这个高级功能，这个功能在1.7的版本中才有，但是我们需要重新编译才能在老版本中使用这个功能。

Flume企业架构

数据仓库模型

扇入

（1）概念：将多个flume agent采集到的数据，全部传递给某一个flume collect，然collect再将数据传递给HDFS

（2）使用组件

①flume agent

source：exec

channel：mem

sink：avro

②flume collect

source：avro

channel:mem

sink：hdfs

（3）运行

①collect： bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/avro-collect.properties -Dflume.root.logger=INFO,console

②agent： bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/avro-agent.properties -Dflume.root.logger=INFO,console

扇出

（1）概念：将同一份数据源采集到不同的多个目标

（2）需求：将hive的日志文件动态读取采集到hdfs的两个目录中

（3）注意：一个sink必须对应一个channel

（4）运行 bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/sinks.properties -Dflume.root.logger=INFO,console

Windows数据源

（1）搭建NFS环境，将Windows日志目录挂载到Linux上

（2）通过flume直接从Linux中读取Windows的日志文件

Taildir的编译实现

1. 获取源码
2. 修改源码
3. 编译
4. 使用

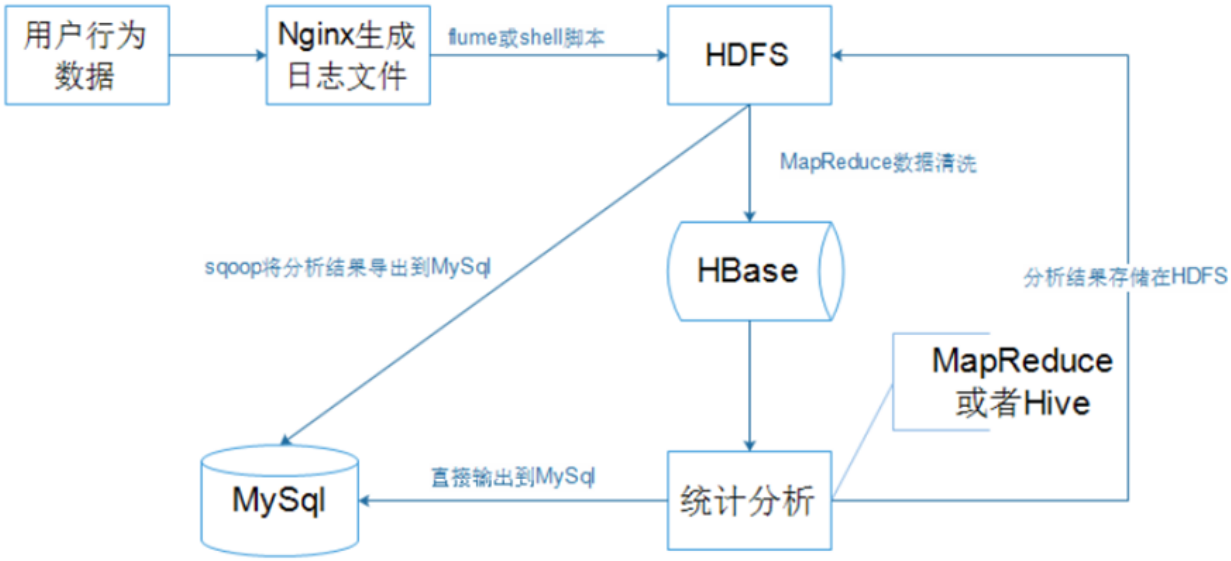
数据采集

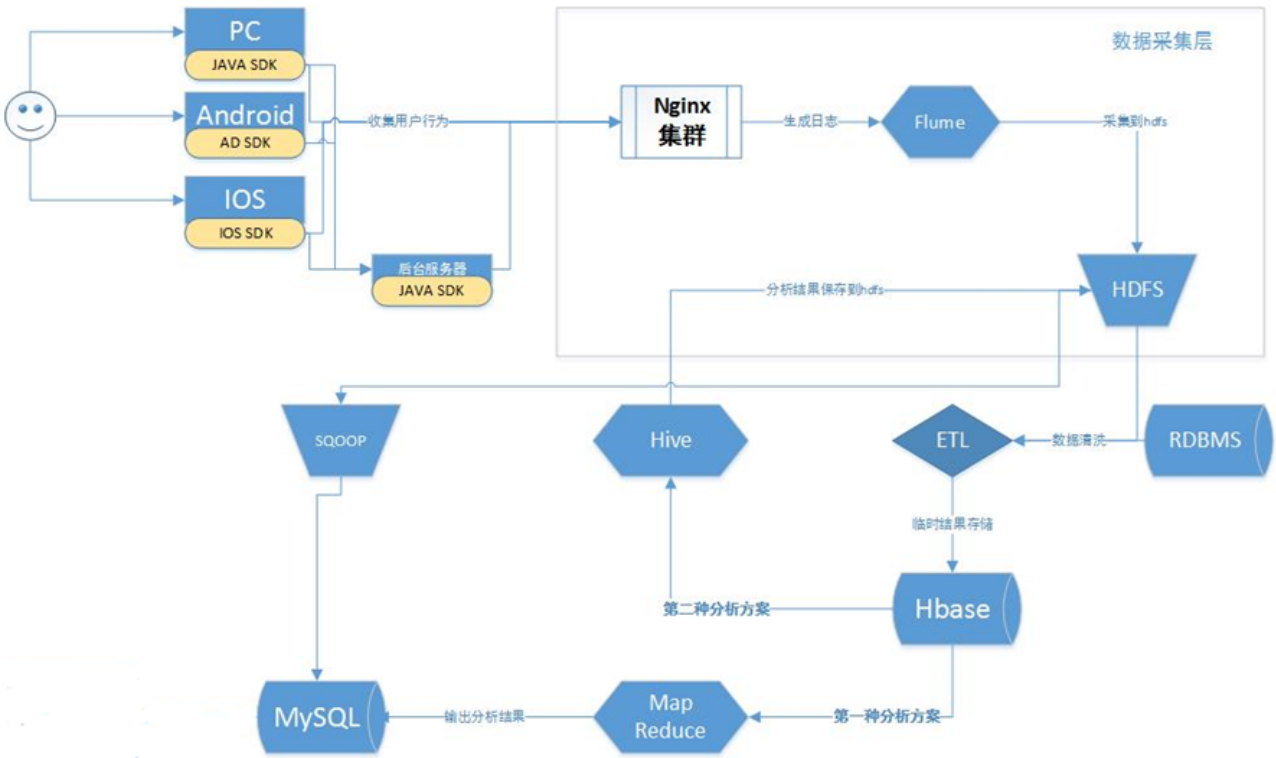
项目技术架构

- Javascript 编写页面日志生成与发送工具

- Java sdk 后台服务日志生成与发送工具
- Nginx 网站服务器，产生日志文件
- Flume 收集日志，导入到HDFS中
- Sqoop ： mysql数据与hdfs,hive,hbase之间数据传输
- HDFS 日志原始数据存储，中间结果存储
- MapReduce 日志数据清洗ETL，批量数据离线分析
- Hive统计分析
- HBase 中间结果存储
- Mysql 最终结果存储，供运营页面查询数据

项目架构图





1. 日志生成及采集层

（1）日志生成：SDK—>Nginx—>日志

（2）日志采集：flume/shell—>HDFS

2. 数据处理分析层

（1）数据分析：MapReduce,Hive

（2）临时结果：Hbase

（3）最终结果：Mysql

3. 数据结果展示层

（1）web项目

（2）springMVC+Highcharts/python 可视化

4. 技术架构选型

（1）Hadoop:hadoop1和hadoop2的区别

①进程

hadoop1:NN,SNN,DN,JT,TT

hadoop2:NN,SNN,DN,RM,NM

②架构区别

hadoop1有单点问题，hadoop2没有

hadoop2有联盟机制Federation

③资源管理

hadoop1中MapReduce负责任务的调度和资源的管理，通过slot进行资源的分配，默认有4个slot，两个map，两个reduce。

hadoop2中yarn负责资源的管理，将所有资源封装成container, application master负责任务的调度。

（2）常见问题：

①hdfs读写

②yarn调度

③mapreduce的原理和过程

（3）Hbase

①表的设计包括rowkey的设计

②hbase和mapreduce的集成

③hbase与hive的集成

④hbase的热点问题

⑤hbase的优化

（4）Hive

①UDF-user-defined function

②数据倾斜

③优化

Nginx的介绍及部署

https://www.cnblogs.com/wcwnina/p/8728391.html

Nginx介绍

nginx是一个高性能的HTTP和反向代理服务器，单台Nginx服务器最多能支持高达50000的并发请求，所以一般情况下，会将Nginx作为静态资源的访问服务器或者作为访问流量分流的服务器

主要特点：占用内存少，并发能力强，扩展容易

http://nginx.org/

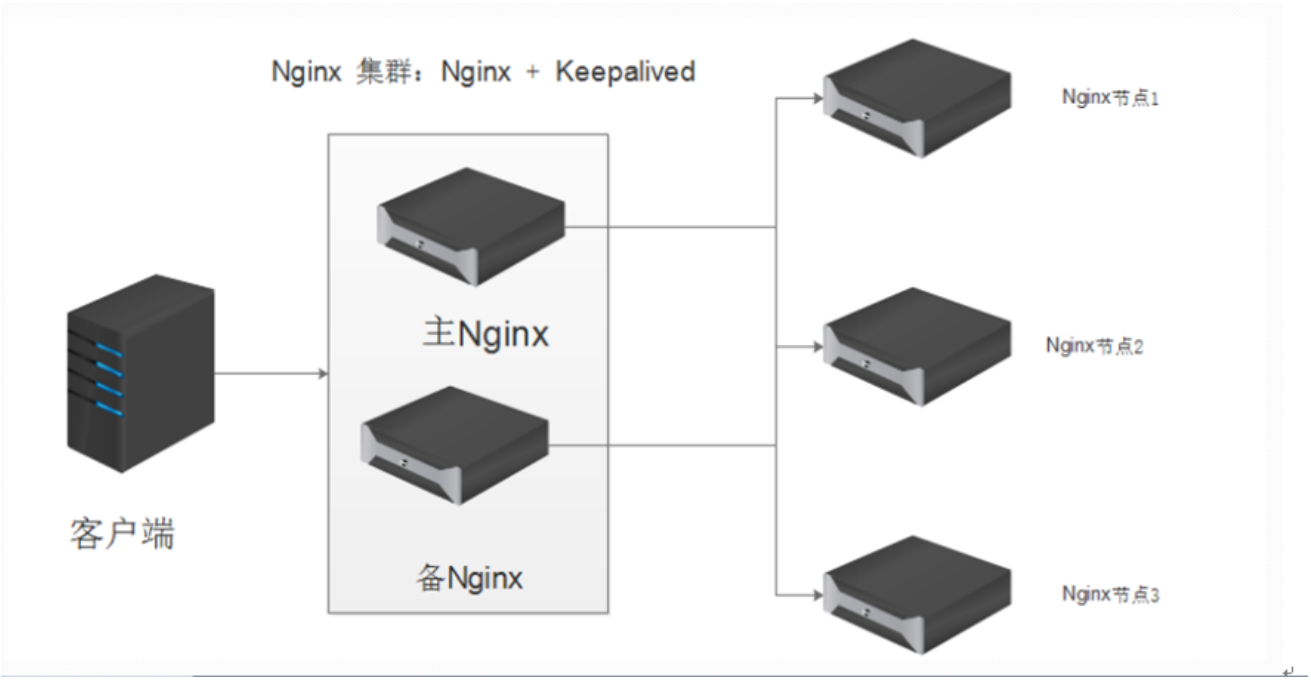
Tengine

在Nginx的基础上，针对大访问量网站的需求，添加了很多高级功能和特性

http://tengine.taobao.org/

Tengine是Nginx的一个重要分支，是淘宝在Nginx的基础上添加了很多的高级功能和特性

NGINX负载均衡模式部署



前端高可用分流：

分流规则：时间轮换/IP地址hash取摸

Nginx安装

数据收集SDK模块

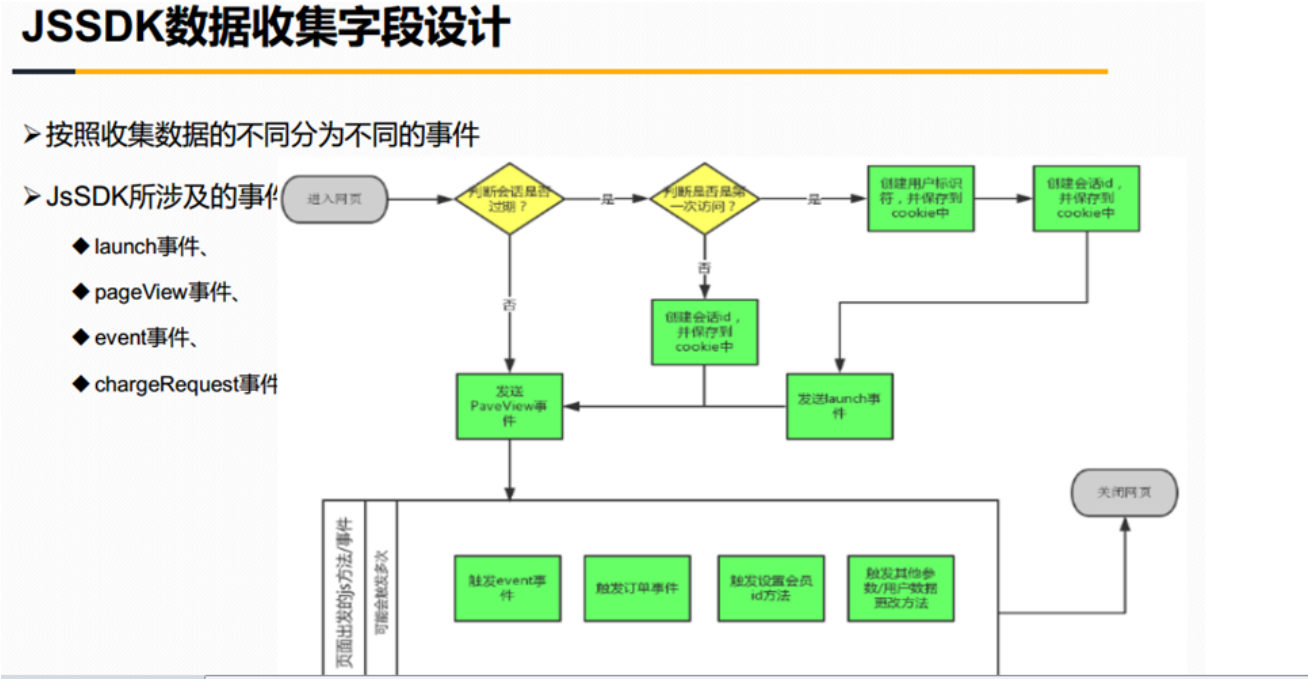
数据收集方案与实现

- 日志数据生成与发送工具设计
- Nginx日志文件生成

- 日志数据导入HDFS

数据收集

- 针对不同的前端，编写对应的SDK模块，比如Java SDK,AndroidSDK,IOS SDK等
- SDK，采用监听事件的方式，对感兴趣的用户行为进行监听，一旦触发，就收集相应的数据
- 数据收集开发的目标：收集更多的数据，减少数据的丢失，减轻对业务系统的侵入



SDK设计

1. 尽量多的收集数据，减少丢失率
2. 减少对业务系统的侵入

常见的事件类型

1. launch事件：第一次进入网站，就会触发，表示用户第一次访问网站的事件类型
2. pageview：浏览页面事件，只要打开任意一个网页就会触发，描述用户访问网站信息，应用于基本的各个不同计算任务
3. event事件：专门记录用户对于某些特定事件/活动的触发行为，主要用于计算各活动的活跃用户以及各个不同访问链路的转化率情况等任务
4. chargerequest事件：该事件的主要作用是记录用户产生订单的行为/数据，为统计计算订单相关的统计结果提供基础数据

JS SDK

1. 创建一个web项目实现用JS SDK/JAVA SDK收集数据
2. demo.html，采用了第一种引用js的方式，访问此页面会触发launch和pageview事件
3. demo2.html,触发chargeRequest事件
4. demo3.html,触发带/不带map和duration的事件
5. demo4.html,采用第二种引用js的方式
6. analytics.js，一旦触发这个js后，就会判断事件类型，然后执行事件中的方法，封装获取到的数据，通过sendDataToServer方法发送给nginx

配置自定义日志格式

1. 修改nginx配置文件　　conf/nginx.conf

①自定义日志格式　　$remote_addr^A$msec^A$http_host^A$request_uri

-客户端的IP- remote_addr

-服务器系统的时间- msec

-访问的主机名- http_host

-请求的uri- request_uri

②修改nginx的主机名：server_name bigdata-1;

③自定义访问资源，凡是访问/computer.jpg的请求，都会采取bigdata的日志格式把日志存放到/export/data/nginx/user_log/access.log，

```
location =/computer.jpg {
    default_type image/jpg;
    access_log /export/data/nginx/user_log/access.log bigdata;
    root /export/data/nginx/html;
}
```

说明：/export/data/nginx/user_log/access.log这个文件必须是nginx自己创建，不能手动创建。

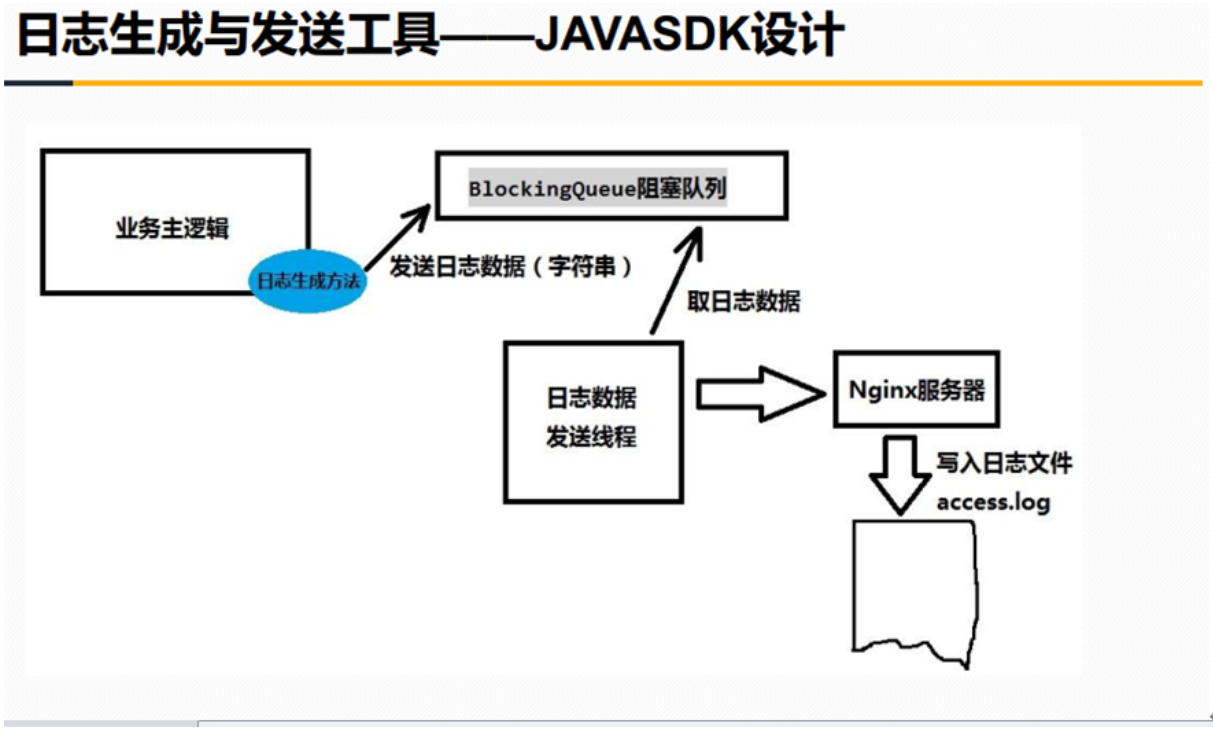
④上传computer.jpg到/export/data/nginx/html

⑤重启nginx服务，发现access.log已经被创建

⑥测试，访问　　发现access.log已经收集到了我们访问静态资源的日志信息：

⑥测试nginx与SDK关联，把web项目发布到tomcat服务器，然后访问demo.html等几个文件，就触发了事件，然后监听把信息发送给nginx，日志的格式是按照我们自定义的日志格式用^A分割的信息。

JAVA SDK



JAVASDK嵌入项目使用

- Javasdk代码很简单，可以打包jar包或者直接拷贝类到具体的项目中
- 正常逻辑处理到Javasdk所关注的事件后，调用JavaSDK提供的api即可
- 相关事件:chargeSuccess事件：订单支付成功事件 chargerefund事件：退款事件

Ngnix生成模拟数据

日志格式

配置接收的字段尽量方便ETL的过程

配置自定义日志格式

1. 修改nginx配置文件 conf/nginx.conf

   ①自定义日志格式 $remote_addr^A$msec^A$http_host^A$request_uri

   -客户端的IP- remote_addr

   -服务器系统的时间- msec

   -访问的主机名- http_host

   -请求的uri- request_uri

   ②修改nginx的主机名：server_name bigdata-1;

   ③自定义访问资源，凡是访问/computer.jpg的请求，都会采取bigdata的日志格式把日志存放到/export/data/nginx/user_log/access.log，

   location =/computer.jpg {

   default_type image/jpg;

   access_log /export/data/nginx/user_log/access.log bigdata;

   root /export/data/nginx/html;

   }

   说明：/export/data/nginx/user_log/access.log这个文件必须是nginx自己创建，不能手动创建。

   ④上传computer.jpg到/export/data/nginx/html

   ⑤重启nginx服务，发现access.log已经被创建

   ⑥测试，访问 发现access.log已经收集到了我们访问静态资源的日志信息：

   ⑥测试nginx与SDK关联，把web项目发布到tomcat服务器，然后访问demo.html等几个文件，就触发了事件，然后监听把信息发送给nginx，日志的格式是按照我们自定义的日志格式用^A分割的信息。

Flume实现日志采集

对于flume的原理其实很容易理解，我们更应该掌握flume的具体使用方法，flume提供了大量内置的Source、Channel和Sink类型。而且不同类型的Source、Channel和Sink可以自由组合——组合方式基于用户设置的配置文件，非常灵活。比如：Channel可以把事件暂存在内存里，也可以持久化到本地硬盘上。Sink可以把日志写入HDFS, HBase，甚至是另外一个Source等等。下面我将用具体的案例详述flume的具体用法。

日志数据导入HDFS

把日志数据上传到HDFS中进行处理，可以分为以下几种情况：

- 如果是日志服务器数据较小，压力较小，可以直接使用shell命令把数据上传到HDFS中；
- 如果是日志服务器数据较大，压力较大，使用NFS在其它一台服务器上上传数据；
- 如果日志服务器非常多，数据量大，使用flume进行数据处理；

测试日志上传

1. flume日志上传

   ①配置采集方案loadLog2HDFS.conf

   ②运行， bin/flume-ng agent --conf conf/ --name a1 --conf-file conf/loadLog2HDFS.conf -Dflume.root.logger=INFO,console

2. 企业中使用脚本在后台运行flume_collector.sh，

flume中特殊组件：sinkgroup

- 负载均衡：load_balance

①功能：避免某一个sink的负载过高，使用多个sink来均衡负载

②决定选举的方式 processor.selector

随机方式 轮询

- 故障转移：failover

①功能：避免sink在工作时发生意外情况，停止工作，影响业务，可以配置， 故障转移，由另外一个backup的sink来接手工作。

②通过配置权重，决定谁是active

a1.sinkgroups.g1.processor.priority.k1 = 5

a1.sinkgroups.g1.processor.priority.k2 = 10

美团的flume使用案例

https://tech.meituan.com/mt_log_system_arch.html

ETL业务分析及实现

脚本实现数据采集

需求

1. nginx默认将所有的日志保存到一个文件中，不会自动按天分割日志文件。
2. 需求：一天一个文件 flume可以按照天进行区分，但是用脚本实现一天一个文件的上传

解决方案

1. 修改nginx源码，支持按天存储日志
2. 编写日志切分脚本，实现日志的分离

编写日志切分脚本

1. 获取昨天的日期，常规企业中的日期格式：年-月-日：2017-01-01/年月日20170101 date -d '-1 day' "+%Y%m%d"
2. 对文件进行移动：/export/data/nginx/user_log/access.log
3. 移动变成：/ export/data/nginx/logs/$yesterday/access_$yesterday.log
4. 重新生成nginx日志文件

①不能自己创建文件，nginx只能识别自己创建的文件

②通过平滑的重新加载配置，让nginx自动重新创建日志文件

③获取nginx进程的pid号

④重新生成 kill -USR1 nginx进程的pid号

⑤编写定时任务用crontab每天0点执行切割脚本

00 00 * * * /bin/bash /export/data/nginx/script/cut_nginx_log.sh

⑥ cut_nginx_log.sh

5. 测试切割迁移脚本

①原来的文件access.log

②执行迁移脚本sh -x cut_nginx_log.sh

③到迁移的目录下，迁移成功

④再看下源文件，发现归零了

⑤sdk收集数据

⑥再看源文件

日志上传的脚本

1. 获取昨天的日期
2. 昨天的日志文件---源

   /export/data/nginx/logs/$yesterday/access_$yesterday.log
3. hdfs的目录--目标

   /nginx/$yesterday/access_$yesterday.log
4. 声明Hadoop的使用用户 export HADOOP_USER_NAME=root
5. 上传 hadoop fs -put source target
6. load_to_hdfs.sh
7. 测试上传脚本

   ①查看hdfs

   ②执行上传脚本 sh -x load_to_hdfs.sh

   ③查看结果

####

ETL实现思路分析

数据清洗过程（ETL）,通过MapReduce实现

1. 字段的提取过滤
2. 字段的格式化
3. 字段的补全

ETL-提取字段

1. 将数据文件 20171220.log中每个字段进行提取，取一条数据---，在hdfs上面创建20171220这一天的文件夹，然后将数据上传到这个文件夹，这个数据只有pv和launch事件，我们对这个文件进行hourly分析。
2. 以^A进行分割

   ①ip ②时间戳 ③主机名 ④URI
3. 对URL进行解析，以？进行分割

   ①请求的资源

   ②事件所收集的所有字段
4. 对事件进行解析，以&进行切割

   ①每一条记录的每一个字段的名称和值
5. 对key=value进行分割，以=进行分割

   ①字段的名称 ②字段的值 ③对某些字段进行解码
6. 通过ip地址分析

   ①国家 ②省份 ③城市
7. 通过客户端字段分析

   ①浏览器的类型 ②浏览器的版本

   ③操作系统名称 ④操作系统版本
8. 将所有的字段的名称和值，存储到map集合中

ETL-存储到Hbase

1. 为什么将ETL后的数据存入hbase?

   ①正常来说，ETL后的数据都是存入HDFS

   ②优点：

   用户的行为数据，所产生的事件信息，不同的事件，所收集的字段是不同的，hbase的存储是按列存储，适合schema是动态的数据，存储效率更好。

   数据分析时，并不需要所有的字段，只需要提取部分字段

   hdfs:将所有数据加载进入mapreduce,进行字段的过滤

   hbase:提前通过regionserver进行并发过滤，然后将过滤后的数据加载到MapReduce,MapReduce的负载就降低了

   hbase的高效的实时查询功能，可以很好的与MapReduce和hive进行集成
2. MapReduce与hbase集成

   ①从hbase中读数据，map继承tablemapper

   ②往hbase写数据，reduce继承tablereduce

   ③输入输出及MapReduce的初始化，tableMapReduceutil
3. 设计hbase的表

日志解析代码实现

导入项目包

1. 修改配置文件的主机名 src/main/resources

   ① core-site

   ② hbase-site

   ③ transformer-env

2. 创建目录 src/main/extr

实现日志的解析代码

1. 创建日志解析类

   ①在ETL的MapReduce中，读取完hdfs上的数据文件

   ②调用解析类进行字段的解析，将解析后的字段存入map集合返回给MapReduce

   ③在包com.bigdata.offline.analystics.util.etl创建日志解析类LogParse.java来实现解析如下信息：

   （ETL-提取字段）里的内容

2. 企业中的IP解析库

   ①公共库

   纯真IP数据库qqwry.dat

   淘宝的IP数据库

   为了保障服务正常运行，每个用户的访问频率需小于10qps

   ②企业中搭建自己的IP数据库

   基于公共的数据库+淘宝IP

Hbase与MapReduce集成

ETL（MapReduce）的流程

将日志解析存入hbase的表，没有排序/合并/聚合的需求，所以一般的ETL程序没有shuffle和reduce过程

MapReduce

1. 输入->map->shuffle->reduce->输出
2. map输入：

   ①从hdfs上读取昨天的日志

   ②数据类型 key：longwritable:行的偏移量 value：Text:日志文件一行的内容

   ③在hdfs上每天的日志是按照日期命名的单独的文件目录存放，所以要动态的去读取
3. 进行日志的解析，调用LogParser类型进行字段的提取和补全
4. 返回的是map集合
5. 输出到Hbase

   ①输出数据类型（map端）封装PUT

   rowkey：实现rowkey的生成

   列簇 列标签 值
6. shuffle：ETL程序中没有

①分区 ②排序 ③合并

7. reduce：分组，合并，排序：ETL程序中没有

   ①输入 ②输出

8. 驱动类

   ①初始化map和reduce

   ②tableMapReduceutil

9. 需要解决的问题

   ①动态的按照日期读取日志

   ②创建hbase表

   如何设计hbase表 rowkey 列簇

Hbase表设计

rowkey

1. 设计原则

   ①长度原则：10-100，最大不超过64k

   ②基本原则

   ③散列原则:热点问题

   ④唯一原则

2. 项目中的rowkey：s_time_CRC32(u_ud+u_md+event):取8位

   ①这样会导致热点问题

   ②解决办法：按天分表，好处：需求是按天统计分析；提高数据处理速度；历史数据容易归档

列簇

1. 企业规范：名称不要太长，可以区别即可
2. 个数:1-2个
3. 项目中选用一个列簇

需要解决问题

1. 热点问题
2. 按天分表：每天的数据解析后存储到一张表中，那么表名的定义也是一个需求解决的问题。

表的名称

event_logs20171220

ETL业务分析及实现（二）

（1）在com.bigdata.offline.analystics.util.etl创建ETLMapper类

case LAUNCH

case PAGEVIEW

case CHARGEREQUEST

case CHARGESUCCESS

case CHARGEREFUND

case EVENT

ETL的驱动类代码实现

在com.bigdata.offline.analystics.util.etl创建ETLDriver类

如何解决动态的识别昨天的日志文件的路径

1. 参数的方式进行传递

   使用参数指定分析某一天的数据

   参数格式 -d 2017-12-20
2. 如果没给参数，默认分析处理昨天的数据
3. 创建hdfs目录/eventLogs/20171220,然后把数据上传到这个目录

创建hbase表

1. create 'tbname','cf'
2. 表名，按天分表
3. 列簇，info
4. 每次都要创建新表

   ①创建表

   如果表已经存在，提前删除表，然后创建

   如果不存在，直接创建

本地配置MapReduce远行

准备

1. 在windows上解压一个Hadoop，不要有中文路径
2. 修改etc/hadoop/hadoop-env.cmd

   set JAVA_HOME="C:\Program Files\Java\jdk1.8.0_181"
3. 配置本地Hadoop的环境变量
4. 将winutils.exe文件放入Hadoop的bin目录下
5. 代码中必须添加修改后的源码

本地远行

1. pom文件

   true

   false
2. driver

① reduce的初始化

② setconf方法中

3. IP解析库必须指定本地的位置

   private static final String ipFilePath = "ip/qqwry.dat";

4. 重新编译

集群远行测试

1. 修改pom文件
2. drive
3. ip解析库
4. 重新编译
5. 上传远行
6. 配置Hadoop远行hbase程序的环境变量

   export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/export/server/hbase-0.98.6-cdh5.3.6/lib/*

7. 重新远行

   bin/yarn jar /export/data/offline_data_analysis-0.0.1-SNAPSHOT.jar
   com.bigdata.offline.analystics.util.etl.ETLDriver -d 2017-12-20

数据分析的思路及代码实现

新增用户分析实现思路

需求：新增用户的统计分析

1. 新增用户：如何区分新老用户？ 新增用户会触发launch事件 en=e_l
2. 统计 统计新增访客的UUID，进行去重，统计个数

多维度的组合分析

1. 时间维度：天维度
2. 地域维度，浏览器维度，平台维度，操作系统维度
3. 分析结果：

   ①昨天的新增访客的个数：基于时间维度（天）

   ②昨天的某一个浏览器类型下的新增访客的个数：基于时间维度+浏览器类型

   ③昨天的某一个浏览器类型下的某个版本的新增访客的个数

   基于时间维度+浏览器维度

   天+浏览器名称+浏览器版本

4. 项目结果数据库

   ①维度表

   时间维度，平台维度，浏览器维度，

KPI维度（用于区别两类分析，比如打标识，在同一个mapreduce进行两类 分析，结果输到到不同的表中 每一条hbase的数据都要做两类分析）

②结果表

我们对Hbase的表中的一条数据进行两类分析，每一条数据可能会输出多条数据，比如，因为时间维度表中有年、月、周、日二级维度；浏览器维度中有浏览器名称和浏览器版本二级维度，二级维度还要排列组合，所以一条数据分析出多条数据

③在MySQL中创建report数据库 在report数据库运行report.sql

程序分析结果（类似于wordcount）

维度 个数

MapReduce实现过程

1. input：用于从hbase中读取所有事件类型为launch事件的记录

   ①数据源是hbase

   ②在输入中直接进行过滤，将过滤好的数据传递给map

   ③MapReduce与hbase集成，继承tablemapper，到hbase去读数据，在driver类中进行map的初始化

   ④过滤构造维度需要的字段

   uuid：访客id

   s_time：时间

   pl：平台的类型

   version：平台的版本

   browsername：浏览器的名称

   browserversion：浏览器版本

2. map

   ①输入：读入过滤好的数据，一次读一个rowkey的数据

   key： 每一条记录的rowkey

   value： 这条rowkey所对应的所有记录

   ②字段的提取

   ③输出：

   -功能：

   -实现值进行合法过滤

   -构造维度-难度

   -key和value

   -key：构造好得维度

   -value：UUID

   -比如结果

   20171220+website+1 uuid1

   20171220+website+chrom uuid1

20171220+website+1 +chrom uuid1

20171220+website+chrom+1 uuid1

20171220+website+1 +chrom+1 uuid1

3. shuffle

   ①默认的shuffle

   ②输入：map的输出

   ③输出：

   -key：20171220+website+1

   -value：{UUID1，UUID2，uuid3……}

4. reduce

   ①输入：shuffle的输出

   ②输出：

   key：20171220+website+1

   value: 去重后的个数

5. output

   ①默认的输出：hdfs-》文件

   ②MySQL

   -自定义MapReduce的输出类型

数据分析的方式

项目中：数据存储在hbase中->ETL->HBASE->MapReduce->Mysql->ETL->hbase->Hive->HDFS->SQOOP->MySQL

Hbase过滤实现及测试

过滤

1. 过滤所有事件类型为launch事件的记录

   ①字段：en，值是e_l

2. 过滤需要分析的字段

   ①uuid：访客id

   ②s_time：时间

   ③pl：平台的类型

   ④version：平台的版本

   ⑤browsername：浏览器的名称

   ⑥browserversion：浏览器版本

3. 所以，在开发的时候要有一个类（HbaseScanUtil.java）对hbase的数据进行过滤（1）（2），然后调用这个类返回一个scan对象给driver。

新增用户Map类代码实现

新增用户Reduce类代码实现

新增用户驱动类代码实现

Hourly分析