统计分析平台后台详细设计说明文档

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订版本** | **修改描述** | **作者** |
| 2014-06-12 | 0.01 | 初稿 | Ian Guo |
| 2014-11-07 | 0.0.2 | 添加了stat-client，stat-server和db-server的处理流程 | Tomli |

目录

[统计分析平台后台详细设计说明文档 1](#_Toc390344260)

[1. 总体描述 3](#_Toc390344261)

[2. 详细设计 4](#_Toc390344262)

[2.1 stat-client 4](#_Toc390344263)

[2.1.1 工作进程描述 4](#_Toc390344264)

[2.1.2 类图 4](#_Toc390344265)

[2.2 stat-server 4](#_Toc390344266)

[2.2.1工作进程描述 4](#_Toc390344267)

[2.2.2 类图 6](#_Toc390344268)

[2.3 日志上传Hadoop 6](#_Toc390344269)

[2.4 db-server 6](#_Toc390344270)

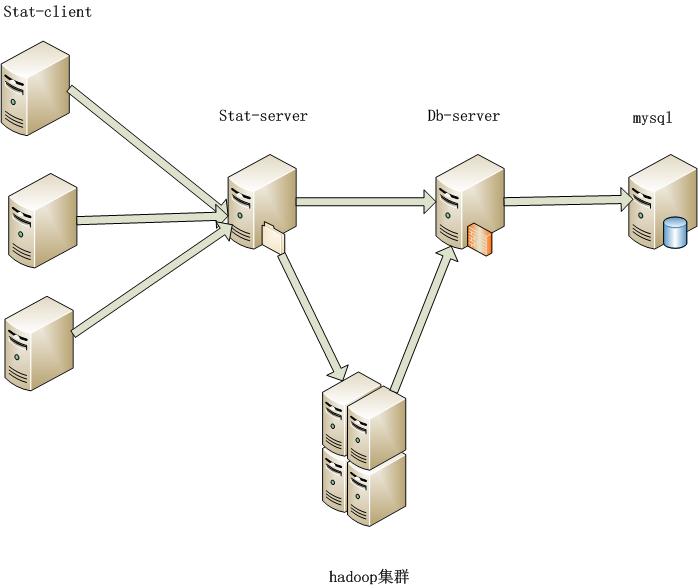
[2.5 config-server 6](#_Toc390344271)

[2.6 stat-center 6](#_Toc390344272)

# 1. 总体描述

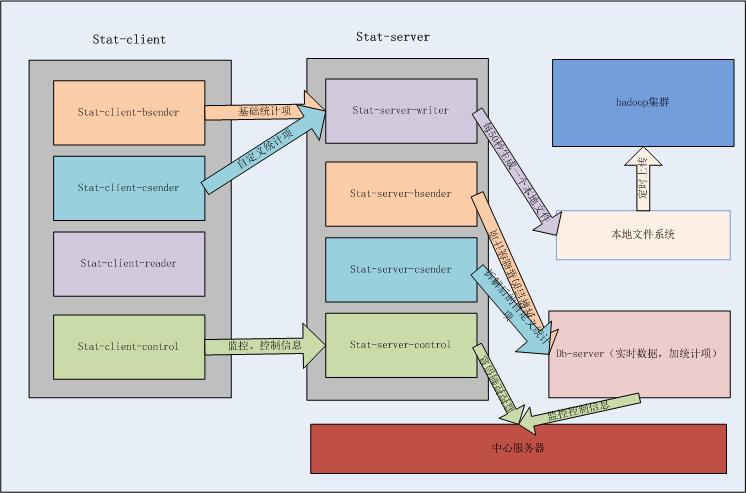
统计分析平台后台（日志收集部分）有多个服务模块：stat-client, stat-server, db-server, config-server, stat-center。

总体部署图（图中config-server未显示）如下：



* stat-client部署在业务机器上，负责将各业务日志文件发送至stat-server。
* stat-server部署在各IDC的出口处，每个IDC一台。负责将stat-client发来的日志合并，写到本地文件，然后解析发送到db-server。另外该机器还部署有日志文件上传脚本，将日志上传到hadoop集群。
* db-server负责实时计算和数据入库。
* stat-center负责平台的监控，报警，升级管理等等。
* hadoop负责离线数据计算。

各服务详细模块如下图：



# 2. 详细设计

## 2.1 stat-client

### 2.1.1 工作进程描述

stat-client有四个工作进程：

* control进程：负责stat-client的控制，监控，管理，比如定时清理sent目录，定时与中心服务器（通过中转）发送heartbeat，定时检查更新，执行更新。
* reader进程：负责将符合发送条件的日志文件从inbox移动到outbox目录。

将inbox目录下的文件名保存到一个map中，对于不合法的文件直接会移动到invalid目录里面，当容器数量达到上线

的时候，则停止往map中添加。

对于上次没有处理完的文件(在reader中只是将文件移动到outbox，在sender里面就是记录上次时间片结束时，发送进行到了哪个文件的偏移量)。

处理完上次未处理完的文件之后，处理map中保存的文件，每处理完一个则将其从map中擦除。

* basic sender进程： 负责发送outbox中基础统计项日志，并将发送完的日志文件移动到sent目录下。

statlog\_sender 类中实现了 process\_statlog 方法，首先调用 statlog\_file\_mmap 类的open 刚才map 中第一个文件，

并将offset 设定为指定的值，这在处理上次未完成的文件时会用到。文件已经打开之后，调用statlog\_file\_mmap类的readline

函数读文件，其首先调用statlog\_file\_mmap 类的statlog\_file\_mmap方法将打开的文件mmap到内存，每次mmap到内存中

的大小可由配置文件中的max\_mmap\_block 来指定，目前此值是 1M，然后找到距离块尾部最近的换行符进行截断。调用

statlog\_sender的sendline将内存中的文件发送出去，在发送前去判断是否是当天的数据，设置协议号：PROTO\_BASIC\_STATLOG

或PROTO\_BASIC\_STATLOG\_OTHER\_DAY，custom同样处理。每次发送的数据块大小由配置文件中的send\_block\_size来决定。

同时statlog\_sender类中的m\_traffic\_control变量用来标识是否打开流量控制，若子类打开的话，那么它每秒钟发送的数据量

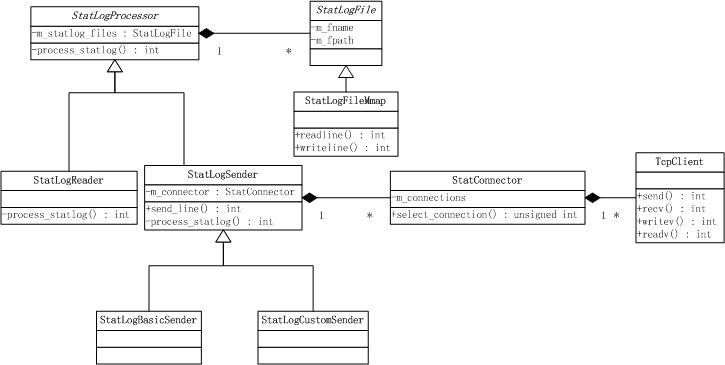
将收到限制，此限制值可由配置文件中的 max\_traffic 来决定，目前大小是4K。线上stat-client的 custom sender 进程就打

开了 m\_traffic\_control 功能。防止发送custom数据占用过大的带宽影响basic数据的发送。发送时会记录发送的数据块大小，

用来做流量监控。

* custom sender进程：负责发送outbox中自定义统计项日志，并将发送完的日志文件移动到sent目录下。该进程发送有流量监控，保证基础统计项优先发送。

### 2.1.2 类图



## 2.2 stat-server

### 2.2.1工作进程描述

框架的作用：

1. select\_channel 根据收到包的协议选择将数据推送给哪个进程
2. 在框架将数据推送给相关进程时，shmq\_pushed会通知stat-client重传，stat-client在收到这种反馈后，将写错误日志，并将其移到发送失败的目录

stat-server有5个工作进程：

* control进程：负责stat-server的控制，监控，管理，比如定时清理log目录，定时与中心服务器发送heartbeat，定时检查更新，执行更新，作为中转，转发stat-client与stat-center的通信数据。
* current writer：负责写当天的日志，将日志文件写在inbox中，固定时间将文件移动到outbox中。

statlog\_cur\_writer\_proc 类 继承了 statlog\_writer\_proc 类，此类用来将从stat-client接受来的数据落地，在

statlog\_writer\_proc类 的 process\_client\_pkg 中会调用 process\_statlog 会根据数据包中的协议号来采用m\_basic\_writer

或者 m\_custom\_writer将接收到的数据分basic和custom落下来，然后执行basic或custom的写操作。

其中 m\_basic\_writer 和 m\_custom\_writer 是由子类 statlog\_cur\_writer\_proc 返回 statlog\_file\_mmap\_writer的对象来对m\_basic\_writer进行初始化。statlog\_file\_writer是statlog\_file\_mmap\_writer的父类，调用 m\_basic\_writer 的 write操作就是调用父类statlog\_file\_writer的write函数，会根据数据包中的gameid和日期来作为索引在map中查找是否有

有对应的文件已经在写，如果有的话就继续写入该文件，如果剩余空间不足，则返回，并将数据msycn落地,将其从map

中去掉；如果没有的话就add\_statlog\_file 生成新的 statlog\_file\_mmap内存文件，即将收到的数据放在内存中。

在 statlog\_writer\_proc类的timer\_event 中会检测距离上次落地的时间间隔是否大于sync-interval, 目前配置文件

中配的是10S， 调用 m\_basic\_writer的sync函数将数据落地，同时会检测map中文件的时长，如果已经超过了

m\_switch\_file\_interval(目前配置文件中是50S) 则会将本文件落地，从而切换新的文件。

再将数据落到inbox目录下之后，再调用rename将文件移动到outbox中。

* other writer：负责写其他时间的日志，用于修复数据、补发日志时使用。将日志文件写在inbox中，固定时间将文件移动到outbox中。

此处用的是stlog\_file\_disk 类来处理从stat-client接收到的数据，直接调用的是 write 操作对数据进行落地，这种

方法效率很低。

* basic sender进程： 负责解析并发送outbox中基础统计项日志，将发送完的日志文件移动到sent目录下。

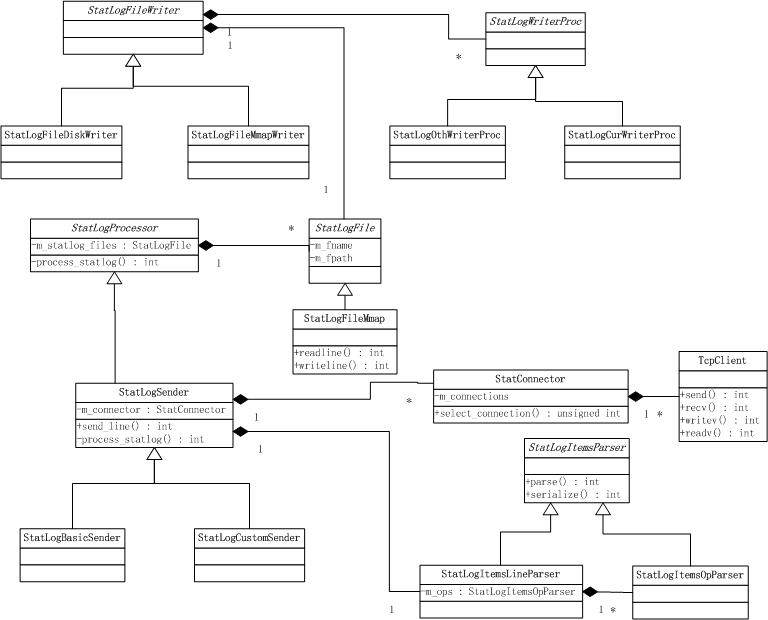
statlog\_sender 是basic\_sender类的父类，statlog\_sender继承了statlog\_processor类，处理过程直接调用

statlog\_processor类的process函数。process会调用statlog\_sender的process\_statlog函数将数据发到db-server。

在将数据发往db-server之前会对日志进行解析，组装协议包发往db-server。

* custom sender进程：负责解析并发送outbox中自定义统计项日志，将发送完的日志文件移动到sent目录下。该进程发送有流量监控，保证基础统计项优先发送。

### 2.2.2 类图



## 2.3 日志上传Hadoop

## 2.4 db-server

### 2.4.1工作进程描述

db-server有2个工作进程：

* control进程：负责db-server的控制，监控，管理，比如定时清理sent目录，定时与中心服务器（通过中转）发送heartbeat，定时检查更新，执行更新。
* db-server进程：负责实时计算和数据入库。

在 process\_client\_pkg 函数中会调用 c\_proto\_pkg 类的recv 函数对接收的数据进行解包(包的格式可见db-server 的相关文档)，从而得到stid，sstid，gid等信息。接下来调用doCmd函数，其中分别调用getGPZSID，getReportId，getDataId等函数，三个函数的执行过程是类似的。首先是从应用程序自己定义的hash\_table缓存中查找，如不存在则从redis中查找，最后向config-server查询，config-server若在redis中查找不到的话，会向数据库中插入。当opType > SET ,是分布操作，先插入report\_id，然后在加工部分会插入data\_id。而opType < SET 时，则先report\_id，紧接着插入data\_id。

db-server的doCmd中对四种协议执行不同的操作，对CMD\_HADOOP\_UPDATE, CMD\_TASK\_UPDATE的数据直接写入数据库，对CMD\_ONLINE\_UPDATE的数据先缓存到一个map中，在timer\_event中定时将数据写入数据库。

## 2.5 config-server

db-server有2个工作进程：

* control进程：负责config-server的控制，监控，管理，比如定时清理sent目录，定时与中心服务器（通过中转）发送heartbeat，定时检查更新，执行更新。
* config-server进程：配合db-server完成统计项的插入

## 2.6 stat-center

### 2.6.1 stat-center功能描述

stat-center只有一个工作进程，负责平台的监控，报警，升级管理。主要功能有：

* 版本管理（升级管理），各服务以安装包、升级包方式存放在固定目录，stat-center自动检查其最新版本。服务模块会定期来查询是否需要升级，如果需要则向stat-center发送下载请求，下载后升级。
* 状态监控，服务启动时，需向stat-center注册，然后定时上报状态信息，状态信息有：硬盘剩余空间大小，inbox,outbox,sent目录文件数等。
* 报警服务分为两种：

1. stat-center根据服务模块发来的状态信息，判断是否需要报警，如需报警，则发送报警。
2. stat-center收到发生错误的信号，发送告警。

### 2.6.2 类图

