Flume 在线采集需求分析与概要设计

# 1.需求分析

采集端Agent监听目录，定时采集数据发送到汇聚端agent。汇聚端按一定的规则将采集文件持久化到本地目录。

需要注意的是：此场景采集的数据都是静态文件，即，如果被采集的文件在采集过程中发生变化（文件更新），可能会导致传输数据错误等问题。

典型场景描述：每天凌晨3点扫描主机host1，/data/logs/目录下的文件，采集新生成的日志文件，发送到汇聚端。

## 采集端

可以制定监听目录采集数据支持:

* 定时扫描目录采集，支持配置unix cron表达式的方式，配置采集定时任务
* 指定序列化方式，默认采用按行序列化的方式
* 支持正则表达式过滤采集文件名
* 断点续传
* 流量控制（限流）

需要自定义开发source以及自定义 interceptor（流量控制）实现上述需求。

## 1.2 接收端

汇聚端即接收端，将接收到的数据持久化到本地目录，支持：

1. 输出文件名与源文件名一致
2. 按主机ip/主机名分类输出归档文件

需要开发自定义sink实现上述需求

# 概要设计

整个采集的流程与模块设计如下图2.1所示：

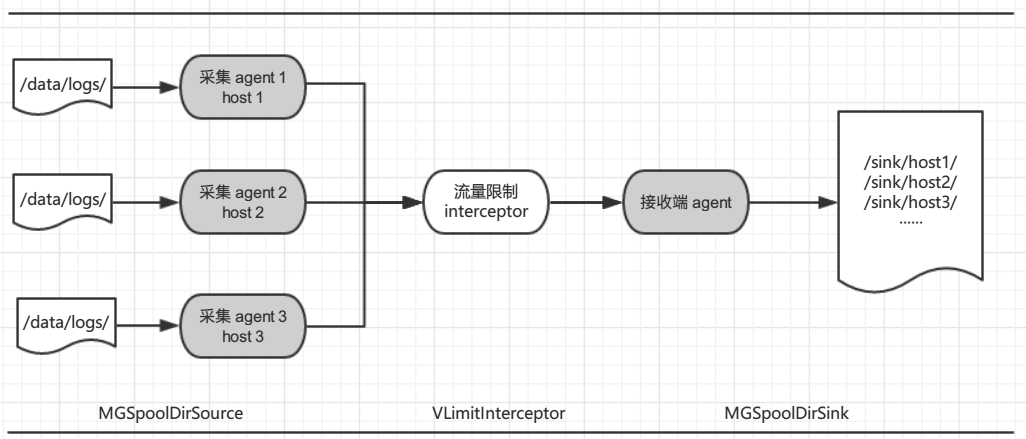


图 2.1 采集架构

主要分为三个模块，采集端自定义source，中间自定义拦截器进行流量限制，接收端自定义sink，自动归档文件到本地。具体使用配置如下：

## 2.1.自定义source – MGSpoolDirSource

自定义source 配置如下：

|  |  |  |
| --- | --- | --- |
| 属性 | 默认值 | 描述 |
| **type** | cn.migu.flume.source.MGSpoolDirSource | 类 |
| **mgSpoolDir** | - | 采集监听的目录 非空 |
| fileHeaderKey | file | 文件的绝对路径 header key |
| basenameHeaderKey | basename | 文件名 header key |
| ignorePattern | ^$ | 文件名过滤规则 |
| trackerDir | .flumespool | 正在传输的文件的track目录 |
| cronExp | 默认为空 | 定时任务表达式 |
| initialDelay | 默认为0，即立刻开始执行采集任务 | 第一次执行时延，**单位：秒** |
| pollNewDelay | 24 ，默认每天重新扫描一次新生成文件 | **单位: 小时** |
| fileDoneHeaderKey | fileDone | 文件传输完成标记 header key: |
| batchSize | 1000 | 每次批量读取events数 |

加粗黑体部分为必须配置的项目，黄色标记部分为定时配置。

### 2.1.1 定时任务配置

支持两种配置定时任务的方式

1. 标准Unix cron expression 配置定时任务
2. 定义初始delay值 ( initialDelay )，与定时循环周期 (pollNewDelay) 的方式

在1，2方式都配置的情况下，第1种 - cron 表达式的优先级更高，如果不配置cron，则以第二种方式的默认值定时执行任务（立即执行采集任务，采集完毕后，24小时后再次扫描）。

### 文件读取规则

当指定监听目录后，读取规则如以下：

1. 默认读取该目录下所有的文件（子目录会跳过）
2. 文件读取顺序优先级：按未读文件最近修改日期， old -> new
3. 根据正则表达式（ignorePattern）过滤需要读取端文件，默认值为^$,即不过滤文件，例如可以配置: ^.\*\.tmp$ ,过滤 .tmp 文件。
4. 过滤已经传输完毕端文件
5. 读取完毕记录已读取文件到本地，同时保证不修改源文件文件名

采集规则流程图如下图2.2所示：

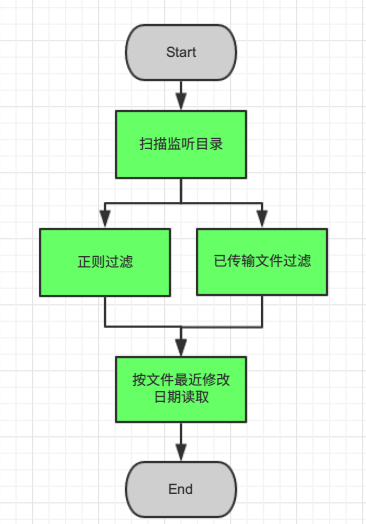


图2.2 文件读取规则

### 2.1.3 文件传输完成标记

设计了flume消息通知机制，当采集的文件全部传输完毕后，发送一条空消息，在消息header 加入采集完毕的标记 fileDone，接收端每次会检测是否收到结束标记，收到即将文件重命名。

## 2.2.自定义sink - MGSpoolFileSink

自定义sink 配置如下：

|  |  |  |
| --- | --- | --- |
| 属性 | 默认值 | 描述 |
| **type** | cn.migu.flume.sink.MGSpoolFileSink |  |
| **sink.directory** | - | 输出目录，非空 |
| hostHeaderKey | hostname | 主机名对应的header key |
| doneFilesTag | fileDone | 文件传输完成header标记 |
| batchEvent | 5000 |  |
| file.rollInterval | 0 | 滚动间隔 ，默认0为不按时间滚动文件 |

doneFilesTag 的配置需要与 source 中的 fileDoneHeaderKey 对应上，目前默认值为：fileDnoe。

## 2.3.自定义interceptor - VLimitInterceptor

自定义interceptor 用来实现流量限制功能，配置如下：

|  |  |  |
| --- | --- | --- |
| 属性 | 默认值 | 描述 |
| **type** | cn.migu.flume.interceptor.VLimitInterceptor |  |
| limitRate | 500 | 限速，默认500Kb/s |
| headerSize | 16 | 头部文件大小：默认16字节 |

该 interceptor配置在source端，即数据发送端，限制每秒数据发送的流量。

## 2.4 agent 监控

监控方案如下图2.4所示，红色标识部分即为心跳监控模块。

在采集agent增加心跳source配置，同时在接受端配置对应的心跳sink，接收心跳信息，并将心跳数据写入MySQL。在web端即可读取MySQL可视化监控信息。

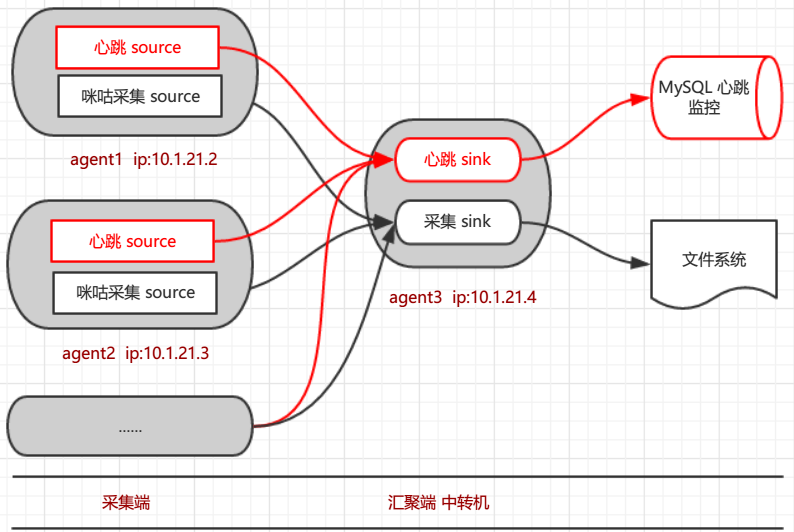


图2.3 Flume agent心跳检测方案

心跳监控source发送的数据如下表所示：

|  |  |  |
| --- | --- | --- |
| 数据 | 类型 | 描述 |
| ip | String | 监控的主机IP |
| agent\_name | String | Agent名 |
| interval\_seconds | Long | 心跳时间间隔 秒为单位 |
| start\_time | Long | 启动时间 时间戳 |

# 部署方式

将项目打包，将以下jar包拷贝至flume lib 目录下即可。

* flume-source-sink-1.0-SNAPSHOT.jar
* cron4j-2.2.5.jar
* cron-utils-4.1.0.jar