# Flume

## 基本概念和配置和使用方式

### 1.1(★) flume的基本概念和使用方式。

基本概念：

flume是一个分布式、可靠、和高可用的海量日志采集、聚合和传输的系统。

Flume支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。

使用方式：

1. netcat方式+Logger日志显示
2. flume级联方式

### 1.2(★★)flume配置方式，flume集群（问的很详细）

配置：

flume是一个可容错的、健壮的并且非常简单的流式数据框架，他只需要简单配置source、channel以及sink后，编写一条命令就可实时采集数据。 agent的三个组成部分为：source、channel和sink。 source：用于采集数据，source是产生数据流的地方，同时，source会将产生的数据传输到channel上。 channel：连接source和sink，类似队列 sink：从channel收集数据，将数据写到目标源，可以是下一个flume的source，或者是hdfs或者hbase等。

Flume的配置围绕着source、channel、sink叙述，flume的集群是做在agent上的，而非机器上。

#配置一个agent agent的名称可以自定义

#指定agent的 sources，sinks，channels

#分别指定 agent的 sources，sinks，channels 的名称 名称可以自定义

a1.sources=s1

a1.channels=c1

a1.sinks=k1

#配置 source 根据 agent的 sources 的名称来对 source 进行配置

#source 的参数是根据 不同的数据源 配置不同---在文档查找即可

#配置目录 source flume这个文件夹用于存储需要读取的文件

a1.sources.s1.type=spooldir

a1.sources.s1.spoolDir=/home/hadoop/apps/apache-flume-1.8.0-bin/flume

#配置 channel 根据 agent的 channels的名称来对 channels 进行配置

#配置内存 channel

a1.channels.c1.type=memory

#配置 sink 根据 agent的sinks 的名称来对 sinks 进行配置

#配置一个 logger sink

a1.sinks.k1.type=logger

#绑定 特别注意 source的channel 的绑定有 s,sink的 channel的绑定没有 s

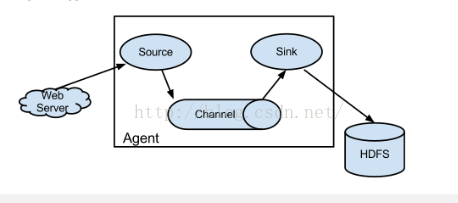
a1.sources.s1.channels=c1

a1.sinks.k1.channel=c1

### 1.3 (★★★)flume使用的时候是二次开发还是原有的配置

使用的是基于源码的二次开发

### 1.4(★)flume的框架是怎么搭建的(画图)？



## 与Kafka的整合

### 2.1 (★)数据怎么采集到Kafka，实现方式。

使用官方提供的flumeKafka插件，插件的实现方式是自定义了flume的sink，将数据从channle中取出，通过kafka的producer写入到kafka中，可以自定义分区等。

### 2. 2(★★)flume 多数据源 是到一个topic？

flume实时收集数据发送到Kafka不同topic。有一个springboot的工程，该工程对外提供一个restful的接口，应用方或其它部门调用该接口推送数据，接收到的数据里面会存放到本地磁盘的一个目录下，一天只会产生一个日志文件（根据业务划分，每一类业务一个文件）。通过使用flume实时采集日志，bizkey表示不同的业务日志的标识，按照bizkey值不同来发送到不同的topic

### 2.3(★)flume拉取数据到kafka中你们是用了几个topic

根据实际情况自由配置

### 2.4(★★)项目中Flume中间为什么用Kafka

系统架构方面的考虑：

（1）如果系统比较简单，应用场景比较单一，从简化系统的角度考虑，在满足应用需求的情况下可能只使用一个比较好。

（2）但是考虑到现有系统业务发展，为了后面的灵活扩展，在先用系统设计时留有一定的扩展性感觉更重要，

(3)可能使用Flume+kafka架构相对只使用Kafka会多占用1-2台机器做Flume日志采集，但是为了方便以后日志数据处理方式的扩展，可以采用Flume+kafka架构。

原理方面的考虑：

个人觉得原理不同，侧重点不同

Flume ：管道流方式，提供了很多的默认实现，让用户通过参数部署，及扩展API.

Kafka ：一个可持久化的分布式的消息队列。

（1）Kafka 是一个非常通用的系统。你可以有许多生产者和很多的消费者共享多个主题Topics。相比之下,Flume是一个专用工具被设计为旨在往HDFS,HBase发送数据。它对HDFS有特殊的优化，并且集成了Hadoop的安全特性。如果数据被多个系统消费的话，使用kafka；如果数据被设计给Hadoop使用，使用Flume。

（2）Flume可以使用拦截器实时处理数据。这些对数据屏蔽或者过量是很有用的。Kafka需要外部的流处理系统才能做到。

（3）Kafka和Flume都是可靠的系统,通过适当的配置能保证零数据丢失。然而，Flume不支持副本事件。于是，如果Flume代理的一个节点奔溃了，即使使用了可靠的文件管道方式，你也将丢失这些事件直到你恢复这些磁盘。如果你需要一个高可靠行的管道，那么使用Kafka是个更好的选择。

## Flume数据来源及处理方式

### 3.1 (★)flume获取数据是用什么方式

flume获取数据有如下方式：

①从网络端口收集数据

②从指定的目录下获取所有文件中的数据

③从kafka消息队列中获取数据

④结合flume-ng-sql-source插件，从传统的关系型数据库中采集数据

### 3.2 (★★)flume的采集处理分别有什么不同

Flume支持在日志系统中定制各类数据发送方，用于收集数据。

Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力 Flume提供了从console（控制台）、RPC（Thrift-RPC）、text（文件）、tail（UNIX tail）、syslog（syslog日志系统，支持TCP和UDP等2种模式），exec（命令执行）等数据源上收集数据的能力。

## Flume采集异常分析

### 1.(★★)flume管道内存，flume宕机了数据丢失怎么解决

1、Flume的channel分为很多种，可以将数据写入到文件

2、防止非首个agent宕机的方法数可以做集群或者主备

### 2.(★★)flume和kafka采集日志区别，采集日志时中间停了，怎么记录之前的日志。

Flume采集日志是通过流的方式直接将日志收集到存储层，而kafka试讲日志缓存在kafka集群，待后期可以采集到存储层。

Flume采集中间停了，可以采用文件的方式记录之前的日志，而kafka是采用offset的方式记录之前的日志。

### 3.(★★)flume的负载均衡和高可用以及数据重复丢失问题（两个人轮流问了一小时，有点懵逼了，高可用忘记了）

负载均衡：source里的event流经channel，进入sink组，在sink组内部根据负载算法（round\_robin、random）选择sink，后续可以选择不同机器上的agent实现负载均衡。

配置如下：

# Name the components on this agent

a1.sources = r1

a1.sinks = k1 k2

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = exec

a1.sources.r1.channels=c1

a1.sources.r1.command=tail -F /home/flume/xx.log

#define sinkgroups

a1.sinkgroups=g1

a1.sinkgroups.g1.sinks=k1 k2

a1.sinkgroups.g1.processor.type=load\_balance

a1.sinkgroups.g1.processor.backoff=true

a1.sinkgroups.g1.processor.selector=round\_robin

#define the sink 1

a1.sinks.k1.type=avro

a1.sinks.k1.hostname=192.168.1.112

a1.sinks.k1.port=9876

#define the sink 2

a1.sinks.k2.type=avro

a1.sinks.k2.hostname=192.168.1.113

a1.sinks.k2.port=9876

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel

a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1

a1.sinks.k2.channel=c1

2）Load balancing Log4J Appender

不同的agent处理同一个client产生的数据。

log4j.rootLogger=INFO,flume

log4j.appender.flume = org.apache.flume.clients.log4jappender.LoadBalancingLog4jAppender

log4j.appender.flume.Hosts = 192.168.1.111:41414 192.168.1.111:41414

高可用：在配置高可用的Flume NG时，需要注意一些事项。在Agent中需要绑定对应的Collector1和Collector2的IP和Port，另外，在配置Collector 节点时，需要修改当前Flume节点的配置文件，Bind的IP（或HostName）为当前节点的IP（或HostName），最后，在启动的时候，指定配置文件中的Agent的Name和配置文件的路径，否则会出错。

数据重复：Flume提供至少一次保证，事件至少被存储一次。有些场景会导致Flume最终会不只一次存储数据。

例如，RPC调用可以设置超时时间，若在超时时间内没有得到相应，及时RPC没有失败，也会被认为失败，从而引发重试。

若RPC没有失败，重试将导致相同事件再次发生。造成这数据存储端数据重复。

## 实际项目开发

### 5.1(★★)flume不采集Nginx日志，通过Logger4j采集日志，优缺点是什么？

优点：Nginx的日志格式是固定的，但是缺少sessionid，通过logger4j采集的日志是带有sessionid的，而session可以通过redis共享，保证了集群日志中的同一session落到不同的tomcat时，sessionId还是一样的，而且logger4j的方式比较稳定，不会宕机。

缺点：不够灵活，logger4j的方式和项目结合过于紧密，而flume的方式比较灵活，拔插式比较好，不会影响项目性能。

## 6. 未处理问题

### 6.1(//TODO) flume拉取文件的时候用了多少rdd

### 6.2(//TODO)flume 的实时和定时数据采集，项目和flume的解耦

### 6.3(//TODO)说说flume，flume的极限是多大（就是event的数量，当时懵逼了，就说了一般设置几十万左右，没测试过）