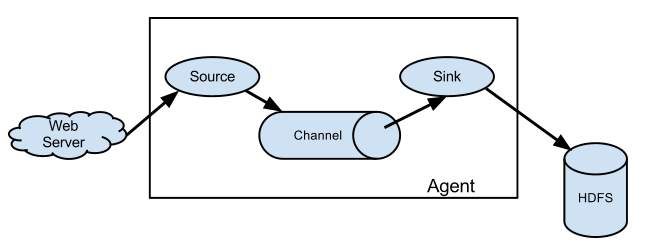
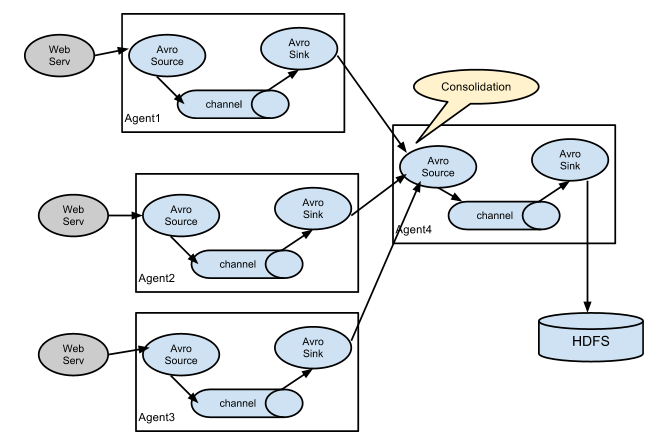
## Flume采集系统结构图

### 简单结构

单个agent采集数据

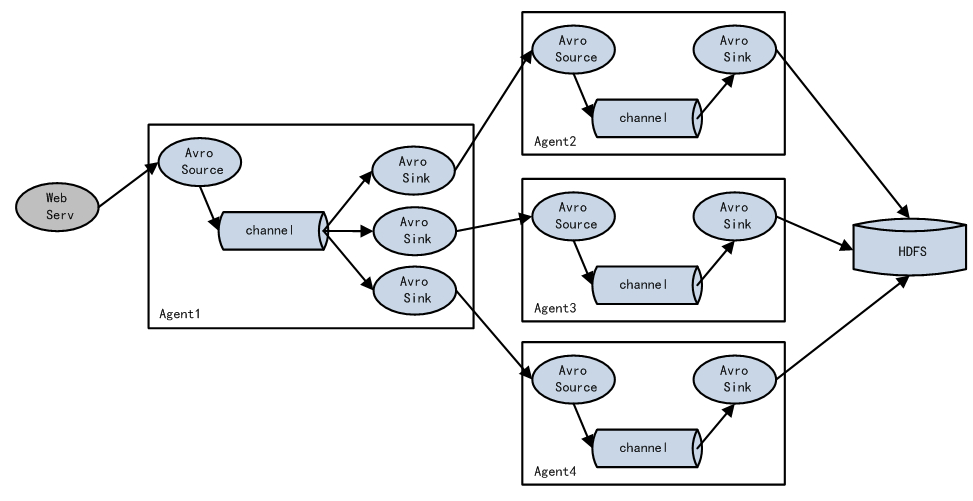
### 复杂结构

多级agent之间串联



# Flume的load-balance、failover

负载均衡是用于解决一台机器(一个进程)无法解决所有请求而产生的一种算法。**Load balancing Sink Processor**能够实现load balance功能，如下图Agent1是一个路由节点，负责将Channel暂存的Event均衡到对应的多个Sink组件上，而每个Sink组件分别连接到一个独立的Agent上，示例配置，如下所示：



|  |
| --- |
| a1.sinkgroups = g1  a1.sinkgroups.g1.sinks = k1 k2 k3  a1.sinkgroups.g1.processor.type = load\_balance  a1.sinkgroups.g1.processor.backoff = true #如果开启，则将失败的sink放入黑名单  a1.sinkgroups.g1.processor.selector = round\_robin # 另外还支持random  a1.sinkgroups.g1.processor.selector.maxTimeOut=10000 #在黑名单放置的超时时间，超时结束时，若仍然无法接收，则超时时间呈指数增长 |

**Failover Sink Processor**能够实现failover功能，具体流程类似load balance，但是内部处理机制与load balance完全不同。

Failover Sink Processor维护一个优先级Sink组件列表，只要有一个Sink组件可用，Event就被传递到下一个组件。故障转移机制的作用是将失败的Sink降级到一个池，在这些池中它们被分配一个冷却时间，随着故障的连续，在重试之前冷却时间增加。一旦Sink成功发送一个事件，它将恢复到活动池。 Sink具有与之相关的优先级，数量越大，优先级越高。

例如，具有优先级为100的sink在优先级为80的Sink之前被激活。如果在发送事件时汇聚失败，则接下来将尝试下一个具有最高优先级的Sink发送事件。如果没有指定优先级，则根据在配置中指定Sink的顺序来确定优先级。

示例配置如下所示：

|  |
| --- |
| a1.sinkgroups = g1  a1.sinkgroups.g1.sinks = k1 k2 k3  a1.sinkgroups.g1.processor.type = failover  a1.sinkgroups.g1.processor.priority.k1 = 5 #优先级值, 绝对值越大表示优先级越高  a1.sinkgroups.g1.processor.priority.k2 = 7  a1.sinkgroups.g1.processor.priority.k3 = 6  a1.sinkgroups.g1.processor.maxpenalty = 20000 #失败的Sink的最大回退期（millis） |

# Flume安装部署

* Flume的安装非常简单

上传安装包到数据源所在节点上

然后解压 tar -zxvf apache-flume-1.6.0-bin.tar.gz

然后进入flume的目录，修改conf下的flume-env.sh，在里面配置JAVA\_HOME

* 根据数据采集需求**配置采集方案**，描述在配置文件中(文件名可任意自定义)
* **指定采集方案配置文件**，在相应的节点上启动flume agent

先用一个最简单的例子来测试一下程序环境是否正常

1. **先在flume的conf目录下新建一个文件**

vi netcat-logger.conf

|  |
| --- |
| # 定义这个agent中各组件的名字  a1.sources = r1  a1.sinks = k1  a1.channels = c1  # 描述和配置source组件：r1  a1.sources.r1.type = netcat  a1.sources.r1.bind = localhost  a1.sources.r1.port = 44444  # 描述和配置sink组件：k1  a1.sinks.k1.type = logger  # 描述和配置channel组件，此处使用是内存缓存的方式  a1.channels.c1.type = memory  a1.channels.c1.capacity = 1000  a1.channels.c1.transactionCapacity = 100  # 描述和配置source channel sink之间的连接关系  a1.sources.r1.channels = c1  a1.sinks.k1.channel = c1 |

1. **启动agent去采集数据**

|  |
| --- |
| bin/flume-ng agent -c conf -f conf/netcat-logger.conf -n a1 -Dflume.root.logger=INFO,console |

-c conf 指定flume自身的配置文件所在目录

-f conf/netcat-logger.con 指定我们所描述的采集方案

-n a1 指定我们这个agent的名字

1. **测试**

先要往agent采集监听的端口上发送数据，让agent有数据可采。

随便在一个能跟agent节点联网的机器上：

telnet anget-hostname port （telnet localhost 44444）

# Flume简单案例

## 采集目录到HDFS

采集需求：**服务器的某特定目录下，会不断产生新的文件，每当有新文件出现，就需要把文件采集到HDFS中去**

根据需求，首先定义以下3大要素

* 采集源，即source——监控文件目录 : **spooldir**
* 下沉目标，即sink——HDFS文件系统 : **hdfs sink**
* source和sink之间的传递通道——channel，可用file channel 也可以用内存channel

配置文件编写：

|  |
| --- |
| # Name the components on this agent  a1.sources = r1  a1.sinks = k1  a1.channels = c1  # Describe/configure the source  ##注意：不能往监控目中重复丢同名文件  a1.sources.r1.type = spooldir  a1.sources.r1.spoolDir = /root/logs  a1.sources.r1.fileHeader = true  # Describe the sink  a1.sinks.k1.type = hdfs  a1.sinks.k1.hdfs.path = /flume/events/%y-%m-%d/%H%M/  a1.sinks.k1.hdfs.filePrefix = events-  a1.sinks.k1.hdfs.round = true  a1.sinks.k1.hdfs.roundValue = 10  a1.sinks.k1.hdfs.roundUnit = minute  a1.sinks.k1.hdfs.rollInterval = 3  a1.sinks.k1.hdfs.rollSize = 20  a1.sinks.k1.hdfs.rollCount = 5  a1.sinks.k1.hdfs.batchSize = 1  a1.sinks.k1.hdfs.useLocalTimeStamp = true  #生成的文件类型，默认是Sequencefile，可用DataStream，则为普通文本  a1.sinks.k1.hdfs.fileType = DataStream  # 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 |

Channel参数解释：

capacity：默认该通道中最大的可以存储的event数量

trasactionCapacity：每次最大可以从source中拿到或者送到sink中的event数量

## 采集文件到HDFS

采集需求：**比如业务系统使用log4j生成的日志，日志内容不断增加，需要把追加到日志文件中的数据实时采集到hdfs**

根据需求，首先定义以下3大要素

* 采集源，即source——监控文件内容更新 : exec ‘tail -F file’
* 下沉目标，即sink——HDFS文件系统 : hdfs sink
* Source和sink之间的传递通道——channel，可用file channel 也可以用 内存channel

配置文件编写：

|  |
| --- |
| # Name the components on this agent  a1.sources = r1  a1.sinks = k1  a1.channels = c1  # Describe/configure the source  a1.sources.r1.type = exec  a1.sources.r1.command = tail -F /root/logs/test.log  a1.sources.r1.channels = c1  # Describe the sink  a1.sinks.k1.type = hdfs  a1.sinks.k1.hdfs.path = /flume/tailout/%y-%m-%d/%H%M/  a1.sinks.k1.hdfs.filePrefix = events-  a1.sinks.k1.hdfs.round = true  a1.sinks.k1.hdfs.roundValue = 10  a1.sinks.k1.hdfs.roundUnit = minute  a1.sinks.k1.hdfs.rollInterval = 3  a1.sinks.k1.hdfs.rollSize = 20  a1.sinks.k1.hdfs.rollCount = 5  a1.sinks.k1.hdfs.batchSize = 1  a1.sinks.k1.hdfs.useLocalTimeStamp = true  #生成的文件类型，默认是Sequencefile，可用DataStream，则为普通文本  a1.sinks.k1.hdfs.fileType = DataStream  # 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 |

**参数解析：**

* **rollInterval**

默认值：30

hdfs sink间隔多长将临时文件滚动成最终目标文件，单位：秒；

如果设置成0，则表示不根据时间来滚动文件；

注：滚动（roll）指的是，hdfs sink将临时文件重命名成最终目标文件，并新打开一个临时文件来写入数据；

* **rollSize**

默认值：1024

当临时文件达到该大小（单位：bytes）时，滚动成目标文件；

如果设置成0，则表示不根据临时文件大小来滚动文件；

* **rollCount**

默认值：10

当events数据达到该数量时候，将临时文件滚动成目标文件；

如果设置成0，则表示不根据events数据来滚动文件；

* **round**

默认值：false

是否启用时间上的“舍弃”，这里的“舍弃”，类似于“四舍五入”。

* **roundValue**

默认值：1

时间上进行“舍弃”的值；

* **roundUnit**

默认值：seconds

时间上进行“舍弃”的单位，包含：second,minute,hour

示例：

a1.sinks.k1.hdfs.path = /flume/events/%y-%m-%d/%H%M/%S

a1.sinks.k1.hdfs.round = true

a1.sinks.k1.hdfs.roundValue = 10

a1.sinks.k1.hdfs.roundUnit = minute

当时间为2015-10-16 17:38:59时候，hdfs.path依然会被解析为：

/flume/events/20151016/17:30/00

因为设置的是舍弃10分钟内的时间，因此，该目录每10分钟新生成一个。

# Flume实战案例

## 日志的采集和汇总

### 案例场景

A、B两台日志服务机器实时生产日志主要类型为access.log、nginx.log、web.log

**现在要求：**

把A、B 机器中的access.log、nginx.log、web.log 采集汇总到C机器上然后统一收集到hdfs中。

但是在hdfs中要求的目录为：

/source/logs/access/20160101/\*\*

/source/logs/nginx/20160101/\*\*

/source/logs/web/20160101/\*\*

### 场景分析

### 数据流程处理分析

### 功能实现

1. 在服务器A和服务器B上

创建配置文件 exec\_source\_avro\_sink.conf

# Name the components on this agent

a1.sources = r1 r2 r3

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = exec

a1.sources.r1.command = tail -F /root/data/access.log

a1.sources.r1.interceptors = i1

a1.sources.r1.interceptors.i1.type = static

## static拦截器的功能就是往采集到的数据的header中插入自## 己定义的key-value对

a1.sources.r1.interceptors.i1.key = type

a1.sources.r1.interceptors.i1.value = access

a1.sources.r2.type = exec

a1.sources.r2.command = tail -F /root/data/nginx.log

a1.sources.r2.interceptors = i2

a1.sources.r2.interceptors.i2.type = static

a1.sources.r2.interceptors.i2.key = type

a1.sources.r2.interceptors.i2.value = nginx

a1.sources.r3.type = exec

a1.sources.r3.command = tail -F /root/data/web.log

a1.sources.r3.interceptors = i3

a1.sources.r3.interceptors.i3.type = static

a1.sources.r3.interceptors.i3.key = type

a1.sources.r3.interceptors.i3.value = web

# Describe the sink

a1.sinks.k1.type = avro

a1.sinks.k1.hostname = 192.168.200.101

a1.sinks.k1.port = 41414

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 20000

a1.channels.c1.transactionCapacity = 10000

# Bind the source and sink to the channel

a1.sources.r1.channels = c1

a1.sources.r2.channels = c1

a1.sources.r3.channels = c1

a1.sinks.k1.channel = c1

1. 在服务器C上创建配置文件 avro\_source\_hdfs\_sink.conf 文件内容为

#定义agent名， source、channel、sink的名称

a1.sources = r1

a1.sinks = k1

a1.channels = c1

#定义source

a1.sources.r1.type = avro

a1.sources.r1.bind = mini2

a1.sources.r1.port =41414

#添加时间拦截器

a1.sources.r1.interceptors = i1

a1.sources.r1.interceptors.i1.type =

org.apache.flume.interceptor.TimestampInterceptor$Builder

#定义channels

a1.channels.c1.type = memory

a1.channels.c1.capacity = 20000

a1.channels.c1.transactionCapacity = 10000

#定义sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.hdfs.path=hdfs://192.168.200.101:9000/source/logs/%{type}/%Y%m%d

a1.sinks.k1.hdfs.filePrefix =events

a1.sinks.k1.hdfs.fileType = DataStream

a1.sinks.k1.hdfs.writeFormat = Text

#时间类型

a1.sinks.k1.hdfs.useLocalTimeStamp = true

#生成的文件不按条数生成

a1.sinks.k1.hdfs.rollCount = 0

#生成的文件按时间生成

a1.sinks.k1.hdfs.rollInterval = 30

#生成的文件按大小生成

a1.sinks.k1.hdfs.rollSize = 10485760

#批量写入hdfs的个数

a1.sinks.k1.hdfs.batchSize = 10000

flume操作hdfs的线程数（包括新建，写入等）

a1.sinks.k1.hdfs.threadsPoolSize=10

#操作hdfs超时时间

a1.sinks.k1.hdfs.callTimeout=30000

#组装source、channel、sink

a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1

1. 配置完成之后，在服务器A和B上的/root/data有数据文件access.log、nginx.log、web.log。先启动服务器C上的flume，启动命令

在flume安装目录下执行 ：

bin/flume-ng agent -c conf -f conf/avro\_source\_hdfs\_sink.conf -name a1 -Dflume.root.logger=DEBUG,console

然后在启动服务器上的A和B，启动命令

在flume安装目录下执行 ：

bin/flume-ng agent -c conf -f conf/exec\_source\_avro\_sink.conf -name a1 -Dflume.root.logger=DEBUG,console

## Flume自定义拦截器（了解）

### 案例背景介绍

Flume是Cloudera提供的一个高可用的，高可靠的，分布式的海量日志采集、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。**Flume有各种自带的拦截器，比如：TimestampInterceptor、HostInterceptor、RegexExtractorInterceptor等，通过使用不同的拦截器，实现不同的功能。**但是以上的这些拦截器，不能改变原有日志数据的内容或者对日志信息添加一定的处理逻辑，当一条日志信息有几十个甚至上百个字段的时候，在传统的Flume处理下，收集到的日志还是会有对应这么多的字段，也不能对你想要的字段进行对应的处理。

### 自定义拦截器

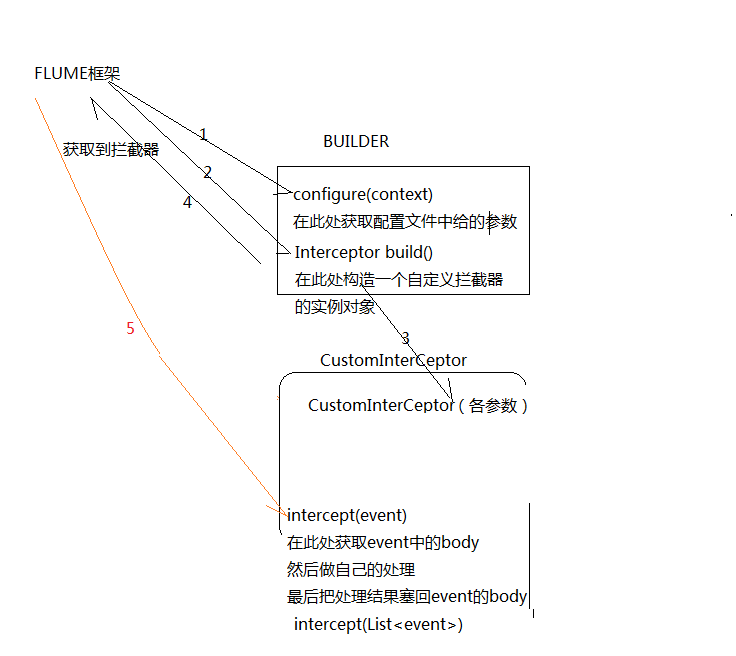
根据实际业务的需求，为了更好的满足数据在应用层的处理，通过自定义Flume拦截器，过滤掉不需要的字段，并对指定字段加密处理，将源数据进行预处理。减少了数据的传输量，降低了存储的开销。

### 功能实现

本技术方案核心包括二部分：

* **编写java代码，自定义拦截器**

内容包括：

1. 定义一个类CustomParameterInterceptor实现Interceptor接口。
2. 在CustomParameterInterceptor类中定义变量，这些变量是需要到 Flume的配置文件中进行配置使用的。每一行字段间的分隔符(fields\_separator)、通过分隔符分隔后，所需要列字段的下标（indexs）、多个下标使用的分隔符（indexs\_separator)、多个下标使用的分隔符（indexs\_separator)。
3. 添加CustomParameterInterceptor的有参构造方法。并对相应的变量进行处理。将配置文件中传过来的unicode编码进行转换为字符串。
4. 写具体的要处理的逻辑intercept()方法，一个是单个处理的，一个是批量处理。
5. 接口中定义了一个内部接口Builder，在configure方法中，进行一些参数配置。并给出，在flume的conf中没配置一些参数时，给出其默认值。通过其builder方法，返回一个CustomParameterInterceptor对象。
6. 定义一个静态类，类中封装MD5加密方法
7. 通过以上步骤，自定义拦截器的代码开发已完成，然后打包成jar， 放到Flume的根目录下的lib中

* **修改Flume的配置信息**

新增配置文件spool-interceptor-hdfs.conf，内容为：

a1.channels = c1

a1.sources = r1

a1.sinks = s1

#channel

a1.channels.c1.type = memory

a1.channels.c1.capacity=100000

a1.channels.c1.transactionCapacity=50000

#source

a1.sources.r1.channels = c1

a1.sources.r1.type = spooldir

a1.sources.r1.spoolDir = /root/data/

a1.sources.r1.batchSize= 50

a1.sources.r1.inputCharset = UTF-8

a1.sources.r1.interceptors =i1 i2

a1.sources.r1.interceptors.i1.type =cn.itcast.interceptor.CustomParameterInterceptor$Builder

a1.sources.r1.interceptors.i1.fields\_separator=\\u0009

a1.sources.r1.interceptors.i1.indexs =0,1,3,5,6

a1.sources.r1.interceptors.i1.indexs\_separator =\\u002c

a1.sources.r1.interceptors.i1.encrypted\_field\_index =0

a1.sources.r1.interceptors.i2.type = org.apache.flume.interceptor.TimestampInterceptor$Builder

#sink

a1.sinks.s1.channel = c1

a1.sinks.s1.type = hdfs

a1.sinks.s1.hdfs.path =hdfs://192.168.200.101:9000/flume/%Y%m%d

a1.sinks.s1.hdfs.filePrefix = event

a1.sinks.s1.hdfs.fileSuffix = .log

a1.sinks.s1.hdfs.rollSize = 10485760

a1.sinks.s1.hdfs.rollInterval =20

a1.sinks.s1.hdfs.rollCount = 0

a1.sinks.s1.hdfs.batchSize = 1500

a1.sinks.s1.hdfs.round = true

a1.sinks.s1.hdfs.roundUnit = minute

a1.sinks.s1.hdfs.threadsPoolSize = 25

a1.sinks.s1.hdfs.useLocalTimeStamp = true

a1.sinks.s1.hdfs.minBlockReplicas = 1

a1.sinks.s1.hdfs.fileType =DataStream

a1.sinks.s1.hdfs.writeFormat = Text

a1.sinks.s1.hdfs.callTimeout = 60000

a1.sinks.s1.hdfs.idleTimeout =60

启动：

bin/flume-ng agent -c conf -f conf/spool-interceptor-hdfs.conf -name a1 -Dflume.root.logger=DEBUG,console

# flume官网:

实际业务架构：

agent串并联 1.多个agent最后sink到一个agent上 --->聚合(consolidation) 2.一个agent最后sink到多个下沉地 (multiplex)

$ bin/flume-ng agent -n $agent\_name -c conf -f conf/flume-conf.properties.template

agent.configuration:

## source:

### Avro Source

type – avro

bind – hostname or IP address to listen on //监听的ip端口

port – Port # to bind to //监听的端口

threads – Maximum number of worker threads to spawn

selector.type //multiplex时配置

selector.\*

interceptors – Space-separated list of interceptors //以空格划分多个拦截器

interceptors.\* //一般填写拦截器的入参 key-value

compression-type none This can be “none” or “deflate”. The compression-type must match the compression-type of matching AvroSource

ssl false Set this to true to enable SSL encryption. You must also specify a “keystore” and a “keystore-password”.

keystore – This is the path to a Java keystore file. Required for SSL.

keystore-password – The password for the Java keystore. Required for SSL.

keystore-type JKS The type of the Java keystore. This can be “JKS” or “PKCS12”.

exclude-protocols SSLv3 Space-separated list of SSL/TLS protocols to exclude. SSLv3 will always be excluded in addition to the protocols specified.

ipFilter false Set this to true to enable ipFiltering for netty //是否开启ip过滤 因为监听的端口有可能多个程序写数据

ipFilterRules – Define N netty ipFilter pattern rules with this config. //IP 过滤的规则

### Exec Source

type – The component type name, needs to be exec

command – The command to execute //tail -F /var/log/secure

shell – A shell invocation used to run the command. e.g. /bin/sh -c. //配置需要什么shell

restartThrottle 10000 Amount of time (in millis) to wait before attempting a restart

restart false Whether the executed cmd should be restarted if it dies //是否重启命令 当命令挂了的时候

logStdErr false Whether the command’s stderr should be logged

batchSize 20 The max number of lines to read and send to the channel at a time //一次读多少行数据和一次向channel发多少条数据

batchTimeout 3000 Amount of time (in milliseconds) to wait, if the buffer size was not reached, before data is pushed downstream

selector.type replicating replicating or multiplexing //multiplex时使用

selector.\* Depends on the selector.type value

interceptors – Space-separated list of interceptors

interceptors.\*

### Spooling Directory Source

type – spooldir

spoolDir – The directory from which to read files from. //配置监控的文件夹

fileSuffix .COMPLETED Suffix to append to completely ingested files //读完一个文件后添加的后缀

deletePolicy never When to delete completed files: never or immediate //读完一个文件后是否删除文件

fileHeader false Whether to add a header storing the absolute path filename.//是否向event的header里添加文件的绝对路径

fileHeaderKey file Header key to use when appending absolute path filename to event header.//文件绝对路径的key

basenameHeader false Whether to add a header storing the basename of the file. //basename: /usr/abc.txt ---->abc

basenameHeaderKey basename Header Key to use when appending basename of file to event header. //basename的key

includePattern ^.\*$ Regular expression specifying which files to include. It can used together with ignorePattern. If a file matches both ignorePattern and includePattern regex, the file is ignored.

ignorePattern ^$ Regular expression specifying which files to ignore (skip). It can used together with includePattern. If a file matches both ignorePattern and includePattern regex, the file is ignored.

trackerDir .flumespool Directory to store metadata related to processing of files.

If this path is not an absolute path, then it is interpreted as relative to the spoolDir. //记录文件的读取进度的文件夹

consumeOrder oldest In which order files in the spooling directory will be consumed oldest, youngest and random.

In case of oldest and youngest, the last modified time of the files will be used to compare the files.

In case of a tie, the file with smallest lexicographical order will be consumed first.

In case of random any file will be picked randomly.

When using oldest and youngest the whole directory will be scanned to pick the oldest/youngest file, which might be slow if there are a large number of files,

while using random may cause old files to be consumed very late if new files keep coming in the spooling directory. //消费文件的次序 以文件最后修改的时间为准

pollDelay 500 Delay (in milliseconds) used when polling for new files.

recursiveDirectorySearch false Whether to monitor sub directories for new files to read.// 是否递归子文件夹中的文件

maxBackoff 4000 The maximum time (in millis) to wait between consecutive attempts to write to the channel(s) if the channel is full. The source will start at a low backoff and increase it exponentially each time the channel throws a ChannelException, upto the value specified by this parameter.

batchSize 100 Granularity at which to batch transfer to the channel //一次向chanel中写的数据条数

inputCharset UTF-8 Character set used by deserializers that treat the input file as text.//解析文件的编码

decodeErrorPolicy FAIL What to do when we see a non-decodable character in the input file. FAIL: Throw an exception and fail to parse the file. REPLACE: Replace the unparseable character with the “replacement character” char, typically Unicode U+FFFD. IGNORE: Drop the unparseable character sequence.

deserializer LINE Specify the deserializer used to parse the file into events. Defaults to parsing each line as an event. The class specified must implement EventDeserializer.Builder.

deserializer.\* Varies per event deserializer.

deserializer.maxLineLength 2048 Maximum number of characters to include in a single event. If a line exceeds this length, it is truncated, and the remaining characters on the line will appear in a subsequent event.

deserializer.outputCharset UTF-8 Charset to use for encoding events put into the channel.

selector.type replicating replicating or multiplexing

selector.\* Depends on the selector.type value

interceptors – Space-separated list of interceptors

interceptors.\*

Taildir Source

### type–TAILDIR

filegroups – Space-separated list of file groups. Each file group indicates a set of files to be tailed.//配置多个文件组 每个组对应一个监控的source

filegroups.<filegroupName> – Absolute path of the file group. Regular expression (and not file system patterns) can be used for filename only.

positionFile ~/.flume/taildir\_position.json File in JSON format to record the inode, the absolute path and the last position of each tailing file.

headers.<filegroupName>.<headerKey> – Header value which is the set with header key. Multiple headers can be specified for one file group.

a1.sources.r1.filegroups = f1 f2

a1.sources.r1.filegroups.f1 = /var/log/test1/example.log

a1.sources.r1.headers.f1.headerKey1 = value1 //headerkey可以自己指定key和value

a1.sources.r1.filegroups.f2 = /var/log/test2/.\*log.\*

a1.sources.r1.headers.f2.headerKey1 = value2

byteOffsetHeader false Whether to add the byte offset of a tailed line to a header called ‘byteoffset’.

skipToEnd false Whether to skip the position to EOF in the case of files not written on the position file.

idleTimeout 120000 Time (ms) to close inactive files. If the closed file is appended new lines to, this source will automatically re-open it.//重新查看文件是否有记录添加的间隔时间

writePosInterval 3000 Interval time (ms) to write the last position of each file on the position file.

batchSize 100 Max number of lines to read and send to the channel at a time. Using the default is usually fine.

backoffSleepIncrement 1000 The increment for time delay before reattempting to poll for new data, when the last attempt did not find any new data.

//当读完一个文件后 每次去检查该文件是否有新内容的每次增加时间

maxBackoffSleep 5000 The max time delay between each reattempt to poll for new data, when the last attempt did not find any new data.

//当读完一个文件后 下一次检查该文件的最大间隔时间

cachePatternMatching true Listing directories and applying the filename regex pattern may be time consuming for directories containing thousands of files. Caching the list of matching files can improve performance. The order in which files are consumed will also be cached. Requires that the file system keeps track of modification times with at least a 1-second granularity.

fileHeader false Whether to add a header storing the absolute path filename.

fileHeaderKey file Header key to use when appending absolute path filename to event header.

### Kafka Source

type – org.apache.flume.source.kafka.KafkaSource

kafka.bootstrap.servers – List of brokers in the Kafka cluster used by the source //broker的列表，用来寻找kafka

kafka.consumer.group.id flume Unique identified of consumer group. Setting the same id in multiple sources or agents indicates that they are part of the same consumer group

kafka消费者的组id

kafka.topics – Comma-separated list of topics the kafka consumer will read messages from.//kafka 消费的topics

kafka.topics.regex – Regex that defines set of topics the source is subscribed on. This property has higher priority than kafka.topics and overrides kafka.topics if exists.

batchSize 1000 Maximum number of messages written to Channel in one batch

batchDurationMillis 1000 Maximum time (in ms) before a batch will be written to Channel The batch will be written whenever the first of size and time will be reached.

backoffSleepIncrement 1000 Initial and incremental wait time that is triggered when a Kafka Topic appears to be empty. Wait period will reduce aggressive pinging of an empty Kafka Topic. One second is ideal for ingestion use cases but a lower value may be required for low latency operations with interceptors.

maxBackoffSleep 5000 Maximum wait time that is triggered when a Kafka Topic appears to be empty. Five seconds is ideal for ingestion use cases but a lower value may be required for low latency operations with interceptors.

useFlumeEventFormat false By default events are taken as bytes from the Kafka topic directly into the event body. Set to true to read events as the Flume Avro binary format. Used in conjunction with the same property on the KafkaSink or with the parseAsFlumeEvent property on the Kafka Channel this will preserve any Flume headers sent on the producing side.

setTopicHeader true When set to true, stores the topic of the retrieved message into a header, defined by the topicHeader property.

topicHeader topic Defines the name of the header in which to store the name of the topic the message was received from, if the setTopicHeader property is set to true. Care should be taken if combining with the Kafka Sink topicHeader property so as to avoid sending the message back to the same topic in a loop.

migrateZookeeperOffsets true When no Kafka stored offset is found, look up the offsets in Zookeeper and commit them to Kafka. This should be true to support seamless Kafka client migration from older versions of Flume. Once migrated this can be set to false, though that should generally not be required. If no Zookeeper offset is found, the Kafka configuration kafka.consumer.auto.offset.reset defines how offsets are handled. Check Kafka documentation for details

kafka.consumer.security.protocol PLAINTEXT Set to SASL\_PLAINTEXT, SASL\_SSL or SSL if writing to Kafka using some level of security. See below for additional info on secure setup.

more consumer security props If using SASL\_PLAINTEXT, SASL\_SSL or SSL refer to Kafka security for additional properties that need to be set on consumer.

Other Kafka Consumer Properties – These properties are used to configure the Kafka Consumer. Any consumer property supported by Kafka can be used. The only requirement is to prepend the property name with the prefix kafka.consumer. For example: kafka.consumer.auto.offset.reset

tier1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource

tier1.sources.source1.channels = channel1

tier1.sources.source1.batchSize = 5000

tier1.sources.source1.batchDurationMillis = 2000

tier1.sources.source1.kafka.bootstrap.servers = localhost:9092

tier1.sources.source1.kafka.topics = test1, test2

tier1.sources.source1.kafka.consumer.group.id = custom.g.id

# tier1.sources.source1.kafka.topics.regex = ^topic[0-9]$

# the default kafka.consumer.group.id=flume is used

## sink：

### HDFS Sink

支持文本参数：

%{host} Substitute value of event header named “host”. Arbitrary header names are supported.

%t Unix time in milliseconds

%a locale’s short weekday name (Mon, Tue, ...)

%A locale’s full weekday name (Monday, Tuesday, ...)

%b locale’s short month name (Jan, Feb, ...)

%B locale’s long month name (January, February, ...)

%c locale’s date and time (Thu Mar 3 23:05:25 2005)

%d day of month (01)

%e day of month without padding (1)

%D date; same as %m/%d/%y

%H hour (00..23)

%I hour (01..12)

%j day of year (001..366)

%k hour ( 0..23)

%m month (01..12)

%n month without padding (1..12)

%M minute (00..59)

%p locale’s equivalent of am or pm

%s seconds since 1970-01-01 00:00:00 UTC

%S second (00..59)

%y last two digits of year (00..99)

%Y year (2010)

%z +hhmm numeric timezone (for example, -0400)

%[localhost] Substitute the hostname of the host where the agent is running

%[IP] Substitute the IP address of the host where the agent is running

%[FQDN] Substitute the canonical hostname of the host where the agent is running

notes：使用时间时 hdfs.useLocalTimeStamp is set to true

配置参数：

type – hdfs

hdfs.path – HDFS directory path (eg hdfs://namenode/flume/webdata/)

hdfs.filePrefix FlumeData Name prefixed to files created by Flume in hdfs directory//文件名前缀

hdfs.fileSuffix – Suffix to append to file (eg .avro - NOTE: period is not automatically added)//文件名后缀

hdfs.inUsePrefix – Prefix that is used for temporal files that flume actively writes into //flume产生的临时文件的前缀

hdfs.inUseSuffix .tmp Suffix that is used for temporal files that flume actively writes into //flume产生的临时文件的后缀

hdfs.rollInterval 30 Number of seconds to wait before rolling current file (0 = never roll based on time interval) // 多长时间产生一个新文件

hdfs.rollSize 1024 File size to trigger roll, in bytes (0: never roll based on file size) //文件多大的时候滚动一次

hdfs.rollCount 10 Number of events written to file before it rolled (0 = never roll based on number of events)//记录多少条数据的时候滚动一次

hdfs.idleTimeout 0 Timeout after which inactive files get closed (0 = disable automatic closing of idle files)

hdfs.batchSize 100 number of events written to file before it is flushed to HDFS //一次写入文件的条数

hdfs.codeC – Compression codec. one of following : gzip, bzip2, lzo, lzop, snappy //文件压缩格式

hdfs.fileType SequenceFile File format: currently SequenceFile, DataStream or CompressedStream

(1)DataStream will not compress output file and please don’t set codeC //不压缩的文本

(2)CompressedStream requires set hdfs.codeC with an available codeC //进行压缩后保存

hdfs.maxOpenFiles 5000 Allow only this number of open files. If this number is exceeded, the oldest file is closed.

hdfs.minBlockReplicas – Specify minimum number of replicas per HDFS block. If not specified, it comes from the default Hadoop config in the classpath.//文件保存的副本数

hdfs.writeFormat Writable Format for sequence file records. One of Text or Writable. Set to Text before creating data files with Flume, otherwise those files cannot be read by either Apache Impala (incubating) or Apache Hive.

hdfs.callTimeout 10000 Number of milliseconds allowed for HDFS operations, such as open, write, flush, close. This number should be increased if many HDFS timeout operations are occurring.//hdfs对文件操作的时候超时时间

hdfs.threadsPoolSize 10 Number of threads per HDFS sink for HDFS IO ops (open, write, etc.)

hdfs.rollTimerPoolSize 1 Number of threads per HDFS sink for scheduling timed file rolling

hdfs.kerberosPrincipal – Kerberos user principal for accessing secure HDFS

hdfs.kerberosKeytab – Kerberos keytab for accessing secure HDFS

hdfs.proxyUser

hdfs.round false Should the timestamp be rounded down (if true, affects all time based escape sequences except %t)

hdfs.roundValue 1 Rounded down to the highest multiple of this (in the unit configured using hdfs.roundUnit), less than current time.

hdfs.roundUnit second The unit of the round down value - second, minute or hour.

hdfs.timeZone Local Time Name of the timezone that should be used for resolving the directory path, e.g. America/Los\_Angeles.

hdfs.useLocalTimeStamp false Use the local time (instead of the timestamp from the event header) while replacing the escape sequences.

hdfs.closeTries 0 Number of times the sink must try renaming a file, after initiating a close attempt. If set to 1, this sink will not re-try a failed rename (due to, for example, NameNode or DataNode failure), and may leave the file in an open state with a .tmp extension. If set to 0, the sink will try to rename the file until the file is eventually renamed (there is no limit on the number of times it would try). The file may still remain open if the close call fails but the data will be intact and in this case, the file will be closed only after a Flume restart.

hdfs.retryInterval 180 Time in seconds between consecutive attempts to close a file. Each close call costs multiple RPC round-trips to the Namenode, so setting this too low can cause a lot of load on the name node. If set to 0 or less, the sink will not attempt to close the file if the first attempt fails, and may leave the file open or with a ”.tmp” extension.

serializer TEXT Other possible options include avro\_event or the fully-qualified class name of an implementation of the EventSerializer.Builder interface.

serializer.\*

a1.channels = c1

a1.sinks = k1

a1.sinks.k1.type = hdfs

a1.sinks.k1.channel = c1

a1.sinks.k1.hdfs.path = /flume/events/%y-%m-%d/%H%M%S

a1.sinks.k1.hdfs.filePrefix = events-

a1.sinks.k1.hdfs.round = true

a1.sinks.k1.hdfs.roundValue = 10 2018-10-10/103000 2018-10-10/104000

a1.sinks.k1.hdfs.roundUnit = minute

### Hive Sink

支持的文本参数

%{host} Substitute value of event header named “host”. Arbitrary header names are supported.

%t Unix time in milliseconds

%a locale’s short weekday name (Mon, Tue, ...)

%A locale’s full weekday name (Monday, Tuesday, ...)

%b locale’s short month name (Jan, Feb, ...)

%B locale’s long month name (January, February, ...)

%c locale’s date and time (Thu Mar 3 23:05:25 2005)

%d day of month (01)

%D date; same as %m/%d/%y

%H hour (00..23)

%I hour (01..12)

%j day of year (001..366)

%k hour ( 0..23)

%m month (01..12)

%M minute (00..59)

%p locale’s equivalent of am or pm

%s seconds since 1970-01-01 00:00:00 UTC

%S second (00..59)

%y last two digits of year (00..99)

%Y year (2010)

%z +hhmm numeric timezone (for example, -0400)

配置参数

type – hive

hive.metastore – Hive metastore URI (eg thrift://a.b.com:9083 )

hive.database – Hive database name

hive.table – Hive table name

hive.partition – Comma separate list of partition values identifying the partition to write to.

E.g: If the table is partitioned by (continent: string, country :string, time : string) then ‘Asia,India,2014-02-26-01-21’ will indicate continent=Asia,country=India,time=2014-02-26-01-21

hive.txnsPerBatchAsk 100 Hive grants a batch of transactions instead of single transactions to streaming clients like Flume. This setting configures the number of desired transactions per Transaction Batch. Data from all transactions in a single batch end up in a single file. Flume will write a maximum of batchSize events in each transaction in the batch. This setting in conjunction with batchSize provides control over the size of each file. Note that eventually Hive will transparently compact these files into larger files.

heartBeatInterval 240 (In seconds) Interval between consecutive heartbeats sent to Hive to keep unused transactions from expiring. Set this value to 0 to disable heartbeats.

autoCreatePartitions true Flume will automatically create the necessary Hive partitions to stream to

batchSize 15000 Max number of events written to Hive in a single Hive transaction

maxOpenConnections 500 Allow only this number of open connections. If this number is exceeded, the least recently used connection is closed.

callTimeout 10000 (In milliseconds) Timeout for Hive & HDFS I/O operations, such as openTxn, write, commit, abort.

serializer Serializer is responsible for parsing out field from the event and mapping them to columns in the hive table. Choice of serializer depends upon the format of the data in the event. Supported serializers: DELIMITED and JSON

serializer.delimiter , (Type: string) The field delimiter in the incoming data. To use special characters, surround them with double quotes like “\t”

serializer.fieldnames – Eg. ‘time,,ip,message’ indicates the 1st, 3rd and 4th fields in input map to time, ip and message columns in the hive table.

serializer.serdeSeparator Ctrl-A (Type: character) Customizes the separator used by underlying serde.

There can be a gain in efficiency if the fields in serializer.fieldnames are in same order as table columns, the serializer.delimiter is same as the serializer.serdeSeparator and number of fields in serializer.

fieldnames is less than or equal to number of table columns, as the fields in incoming event body do not need to be reordered to match order of table columns.

Use single quotes for special characters like ‘\t’. Ensure input fields do not contain this character.

NOTE: If serializer.delimiter is a single character, preferably set this to the same character

roundUnit minute The unit of the round down value - second, minute or hour.

roundValue 1 Rounded down to the highest multiple of this (in the unit configured using hive.roundUnit), less than current time

timeZone Local Time Name of the timezone that should be used for resolving the escape sequences in partition, e.g. America/Los\_Angeles.

useLocalTimeStamp false Use the local time (instead of the timestamp from the event header) while replacing the escape sequences.

create table weblogs ( id int , msg string )

partitioned by (continent string, country string, time string)

clustered by (id) into 5 buckets

stored as orc;

a1.channels = c1

a1.channels.c1.type = memory

a1.sinks = k1

a1.sinks.k1.type = hive

a1.sinks.k1.channel = c1

a1.sinks.k1.hive.metastore = thrift://127.0.0.1:9083

a1.sinks.k1.hive.database = logsdb

a1.sinks.k1.hive.table = weblogs

a1.sinks.k1.hive.partition = asia,%{country},%y-%m-%d-%H-%M

a1.sinks.k1.useLocalTimeStamp = false

a1.sinks.k1.round = true

a1.sinks.k1.roundValue = 10

a1.sinks.k1.roundUnit = minute

a1.sinks.k1.serializer = DELIMITED

a1.sinks.k1.serializer.delimiter = "\t"

a1.sinks.k1.serializer.serdeSeparator = '\t'

a1.sinks.k1.serializer.fieldnames =id,,msg

### Avro Sink

type – avro.

hostname – The hostname or IP address to bind to.

port – The port # to listen on.

batch-size 100 number of event to batch together for send.

connect-timeout 20000 Amount of time (ms) to allow for the first (handshake) request.

request-timeout 20000 Amount of time (ms) to allow for requests after the first.

reset-connection-interval none Amount of time (s) before the connection to the next hop is reset. This will force the Avro Sink to reconnect to the next hop. This will allow the sink to connect to hosts behind a hardware load-balancer when news hosts are added without having to restart the agent.

compression-type none This can be “none” or “deflate”. The compression-type must match the compression-type of matching AvroSource

compression-level 6 The level of compression to compress event. 0 = no compression and 1-9 is compression. The higher the number the more compression

ssl false Set to true to enable SSL for this AvroSink. When configuring SSL, you can optionally set a “truststore”, “truststore-password”, “truststore-type”, and specify whether to “trust-all-certs”.

trust-all-certs false If this is set to true, SSL server certificates for remote servers (Avro Sources) will not be checked. This should NOT be used in production because it makes it easier for an attacker to execute a man-in-the-middle attack and “listen in” on the encrypted connection.

truststore – The path to a custom Java truststore file. Flume uses the certificate authority information in this file to determine whether the remote Avro Source’s SSL authentication credentials should be trusted. If not specified, the default Java JSSE certificate authority files (typically “jssecacerts” or “cacerts” in the Oracle JRE) will be used.

truststore-password – The password for the specified truststore.

truststore-type JKS The type of the Java truststore. This can be “JKS” or other supported Java truststore type.

exclude-protocols SSLv3 Space-separated list of SSL/TLS protocols to exclude. SSLv3 will always be excluded in addition to the protocols specified.

maxIoWorkers 2 \* the number of available processors in the machine

a1.channels = c1

a1.sinks = k1

a1.sinks.k1.type = avro

a1.sinks.k1.channel = c1

a1.sinks.k1.hostname = 10.10.10.10

a1.sinks.k1.port = 4545

### Kafka Sink

type – org.apache.flume.sink.kafka.KafkaSink

kafka.bootstrap.servers – List of brokers Kafka-Sink will connect to, to get the list of topic partitions This can be a partial list of brokers, but we recommend at least two for HA. The format is comma separated list of hostname:port

kafka.topic default-flume-topic The topic in Kafka to which the messages will be published.

If this parameter is configured, messages will be published to this topic.

If the event header contains a “topic” field, the event will be published to that topic overriding the topic configured here.

Arbitrary header substitution is supported, eg. %{header} is replaced with value of event header named “header”.

(If using the substitution, it is recommended to set “auto.create.topics.enable” property of Kafka broker to true.)

flumeBatchSize 100 How many messages to process in one batch. Larger batches improve throughput while adding latency.

kafka.producer.acks 1 How many replicas must acknowledge a message before its considered successfully written.

Accepted values are

0 (Never wait for acknowledgement),

1 (wait for leader only),

-1 (wait for all replicas) Set this to -1 to avoid data loss in some cases of leader failure.

useFlumeEventFormat false By default events are put as bytes onto the Kafka topic directly from the event body. Set to true to store events as the Flume Avro binary format. Used in conjunction with the same property on the KafkaSource or with the parseAsFlumeEvent property on the Kafka Channel this will preserve any Flume headers for the producing side.

defaultPartitionId – Specifies a Kafka partition ID (integer) for all events in this channel to be sent to, unless overriden by partitionIdHeader. By default, if this property is not set, events will be distributed by the Kafka Producer’s partitioner - including by key if specified (or by a partitioner specified by kafka.partitioner.class).

partitionIdHeader – When set, the sink will take the value of the field named using the value of this property from the event header and send the message to the specified partition of the topic. If the value represents an invalid partition, an EventDeliveryException will be thrown. If the header value is present then this setting overrides defaultPartitionId.

allowTopicOverride true When set, the sink will allow a message to be produced into a topic specified by the topicHeader property (if provided).

topicHeader topic When set in conjunction with allowTopicOverride will produce a message into the value of the header named using the value of this property. Care should be taken when using in conjunction with the Kafka Source topicHeader property to avoid creating a loopback.

kafka.producer.security.protocol PLAINTEXT Set to SASL\_PLAINTEXT, SASL\_SSL or SSL if writing to Kafka using some level of security. See below for additional info on secure setup.

more producer security props If using SASL\_PLAINTEXT, SASL\_SSL or SSL refer to Kafka security for additional properties that need to be set on producer.

Other Kafka Producer Properties – These properties are used to configure the Kafka Producer. Any producer property supported by Kafka can be used. The only requirement is to prepend the property name with the prefix kafka.producer. For example: kafka.producer.linger.ms

a1.sinks.k1.channel = c1

a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink

a1.sinks.k1.kafka.topic = mytopic

a1.sinks.k1.kafka.bootstrap.servers = localhost:9092

a1.sinks.k1.kafka.flumeBatchSize = 20

a1.sinks.k1.kafka.producer.acks = 1

a1.sinks.k1.kafka.producer.linger.ms = 1

a1.sinks.k1.kafka.producer.compression.type = snappy

## Flume Channels

### Memory Channel

capacity 100 The maximum number of events stored in the channel

transactionCapacity 100 The maximum number of events the channel will take from a source or give to a sink per transaction

keep-alive 3 Timeout in seconds for adding or removing an event

byteCapacityBufferPercentage 20 Defines the percent of buffer between byteCapacity and the estimated total size of all events in the channel, to account for data in headers.

byteCapacity see description Maximum total bytes of memory allowed as a sum of all events in this channel.

The implementation only counts the Event body, which is the reason for providing the byteCapacityBufferPercentage configuration parameter as well.

Defaults to a computed value equal to 80% of the maximum memory available to the JVM (i.e. 80% of the -Xmx value passed on the command line).

Note that if you have multiple memory channels on a single JVM, and they happen to hold the same physical events (i.e. if you are using a replicating channel selector from a single source) then those event sizes may be double-counted for channel byteCapacity purposes.

Setting this value to 0 will cause this value to fall back to a hard internal limit of about 200 GB.

a1.channels = c1

a1.channels.c1.type = memory

a1.channels.c1.capacity = 10000 //event 条数

a1.channels.c1.transactionCapacity = 10000

a1.channels.c1.byteCapacityBufferPercentage = 20 //channel中的缓存区大小

a1.channels.c1.byteCapacity = 800000 //channel大小

### Flume Channel Selectors

Replicating Channel Selector (default)

a1.sources = r1

a1.channels = c1 c2 c3

a1.sources.r1.selector.type = replicating

a1.sources.r1.channels = c1 c2 c3

Multiplexing Channel Selector

a1.sources = r1

a1.channels = c1 c2 c3 c4

a1.sources.r1.selector.type = multiplexing

a1.sources.r1.selector.header = state //根据r1中header中的state字段来选择send到哪个channel中

a1.sources.r1.selector.mapping.CZ = c1

a1.sources.r1.selector.mapping.US = c2 c3

a1.sources.r1.selector.default = c4 //没有匹配上的send到C4

flume中有三种可监控文件或目录的source，分别问exec、spooldir、taildir  
exec：可通过tail -f命令去tail住一个文件，然后实时同步日志到sink

tail -f 等同于--follow=descriptor，根据文件描述符进行追踪，当文件改名或被删除，追踪停止（慎用）

tail -F 等同于--follow=name --retry，根据文件名进行追踪，并保持重试，即该文件被删除或改名后，如果再次创建相同的文件名，会继续追踪 适合追踪单个文件。  
spooldir：可监听一个目录，同步目录中的新文件到sink,被同步完的文件可被立即删除或被打上标记。适合用于同步新文件，但不适合对实时追加日志的文件进行监听并同步。（适合追踪有新文件产生的目录，可设置递归追踪子文件夹）  
taildir：可实时监控一批文件，并记录每个文件最新消费位置，agent进程重启后不会有重复消费的问题。

a1.sources.r1.type = TAILDIR

# 元数据位置

a1.sources.r1.positionFile = /home/hadoop/data/bd/taildir\_position.json

# 监控的目录

a1.sources.r1.filegroups = f1 f2

a1.sources.r1.filegroups.f1=/home/hadoop/data/bd1/.\*log

a1.sources.r1.filegroups.f2=/home/hadoop/data/bd2/.\*log