Apache Flume

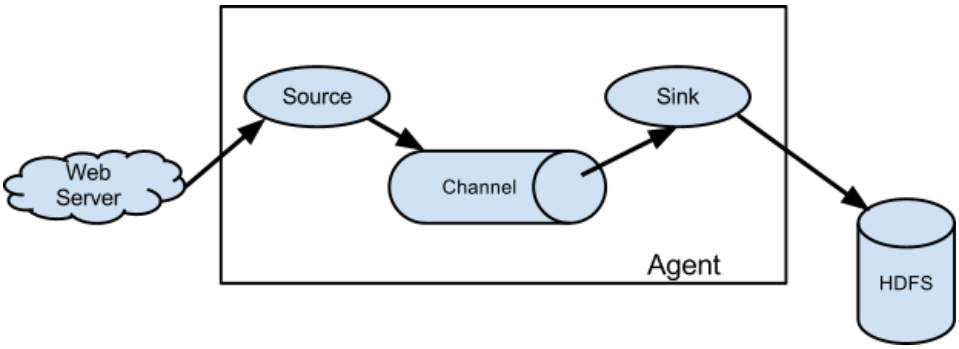
介绍

概要

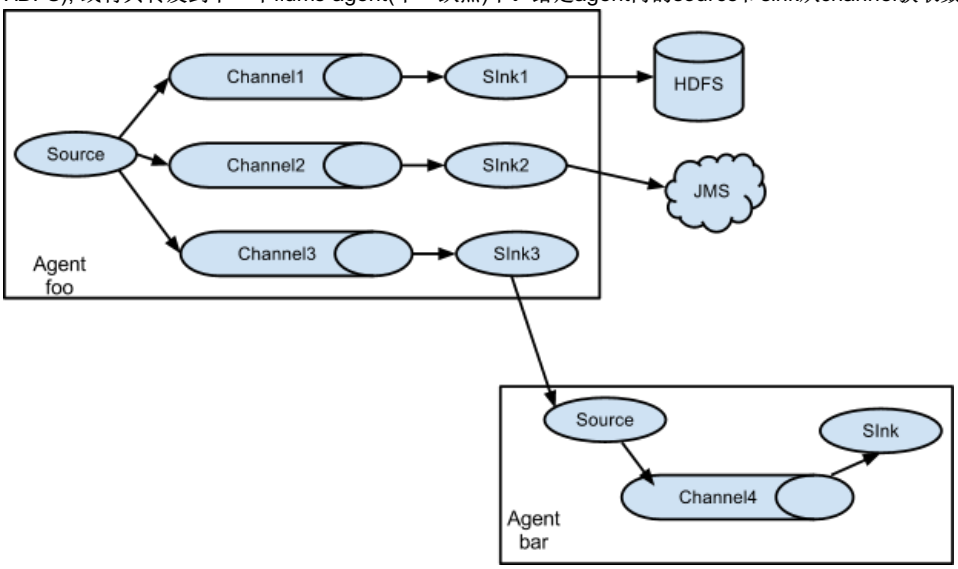
Apache flume是一个分布式的、可靠的、可用的系统, 可以高效地收集、聚合和移动大量不同来源的日志数据到同一个数据存储区。
Apache flume的使用不仅限于日志数据聚合。由于数据源是可定制的, 因此可以用来传输大量的事件数据, 包括但不限于网络流量数据、社交媒体生成的数据、电子邮件以及几乎所有可能的数据源。
Apache flume，现已是 apache 软件基金会的顶级项目。

功能结构

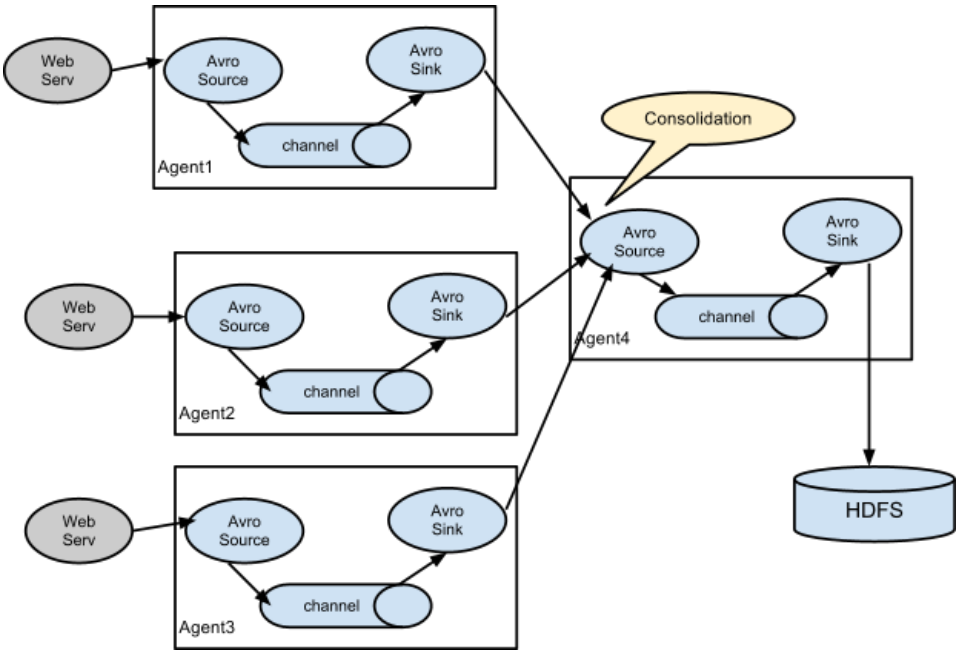
- 数据流模型
  一个flume event被定义为一个数据流单元, 它具有字节有效负载和一组可选的字符串属性。flume agent是一个 (JVM) 进程, 它是作为事件从外部数据源流向下一个目标中转的组件。



  flume使用外部源 (如 web 服务器) 来给它传递事件。外部源以目标flume识别的格式向flume发送event。例如, 一个 Avro 的flume源可以用来接收来自 Avro 客户端或其他flume agent的 Avro event, 从 Avro 接收器发送event。一个类似的流程可以定义使用thrift flume源接从一个thrift flume sink或flume thrift rpc client或任意语言编写的thrift client获取数据。当flume source接收到一个event时, 它会将它存储到一个或多个channel中。该channel被动存储数据, 直到它被flume sink消费掉。file channel-它由本地文件系统支持。sink从channel消费数据, 并将其放入外部存储库 (如 HDFS), 或将其转发到下一个flume agent(下一跃点)中。给定agent内的source和sink从channel获取数据的过程中是异步的。



- 混合数据流
  flume允许用户构建多条数据流, 在这些流中, event在到达最终目的地之前通过多个agent进行移动。它还允许扇入和扇出流、上下文路由和备份路由 (故障转移) 来进行故障agent的数据转移。



- 可靠性
  event在每个agent的channel中存储。然后将event传递到下一个agent或终端存储库 (如 HDFS)。仅在将这些event在下一个agent channel或终端存储库存储后, 才会从channel中删除。 flume使用一种事务性的方法来保证event的可靠交付。source和sink分别在事务中封装由channel提供的事务功能，来放置或提供的event的存储/检索。这可以保证event在端到端之间可靠的传输。在多跳流的情况下, 来自上一个跃点的sink和下一跳的source都有它们的事务运行, 以确保数据安全地存储在下一跳的channel中。
- 可恢复
  event存储在channel中，用来在失败的时候恢复。flume支持由本地文件系统支持的持久file channel。还有一个memory channel, 它简单地将event存储在内存中, 这种速度更快，但当channel所在进程关闭时，数据无法从内存中恢复。

安装使用

1. 下载 Apache Flume
2. 创建example.conf,用来编写flume接收发送的配置
   - 配置格式

```
# list the sources, sinks and channels for the agent
<Agent>.sources = <Source>
<Agent>.sinks = <Sink>
<Agent>.channels = <Channel1> <Channel2>

# set channel for source
<Agent>.sources.<Source>.channels = <Channel1> <Channel2> ...

# set channel for sink
<Agent>.sinks.<Sink>.channel = <Channel1>

# properties for sources
<Agent>.sources.<Source>.<someProperty> = <someValue>

# properties for channels
<Agent>.channel.<Channel>.<someProperty> = <someValue>

# properties for sinks
<Agent>.sources.<Sink>.<someProperty> = <someValue>
```

   - example.conf配置说明:

```
#该配置读取一个文件夹内的文件往es写数据，数据缓存在本地文件中
a1.sources = r1
a1.sinks = k1
```

```
a1.channels = c1

#source为文件夹
a1.sources.r1.type = spooldir
a1.sources.r1.spoolDir = F:/dev/flume/test
a1.sources.r1.channels = c1
a1.sources.r1.deletePolicy = immediate

#sink为elasticsearch
a1.sinks.k1.type = elasticsearch
a1.sinks.k1.hostNames = 192.168.119.215,192.168.119.216,192.168.119.217
a1.sinks.k1.indexName = agent
a1.sinks.k1.indexType = logs
a1.sinks.k1.indexNameBuilder = org.apache.flume.sink.elasticsearch.TimeBasedIndexNameBuilder
a1.sinks.k1.clusterName = elasticsearch-cluster
a1.sinks.k1.batchSize = 500
a1.sinks.k1.ttl = 5d
a1.sinks.k1.channel = c1

#channel为本地文件
a1.channels.c1.type = file
a1.channels.c1.checkpointDir = F:/dev/flume/checkpoint
a1.channels.c1.dataDirs = F:/dev/flume/data

a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

3. 启动一个agent

```
bin/flume-ng agent --conf conf --conf-file example.conf --name a1 -Dflume.root.logger=INFO,console
```

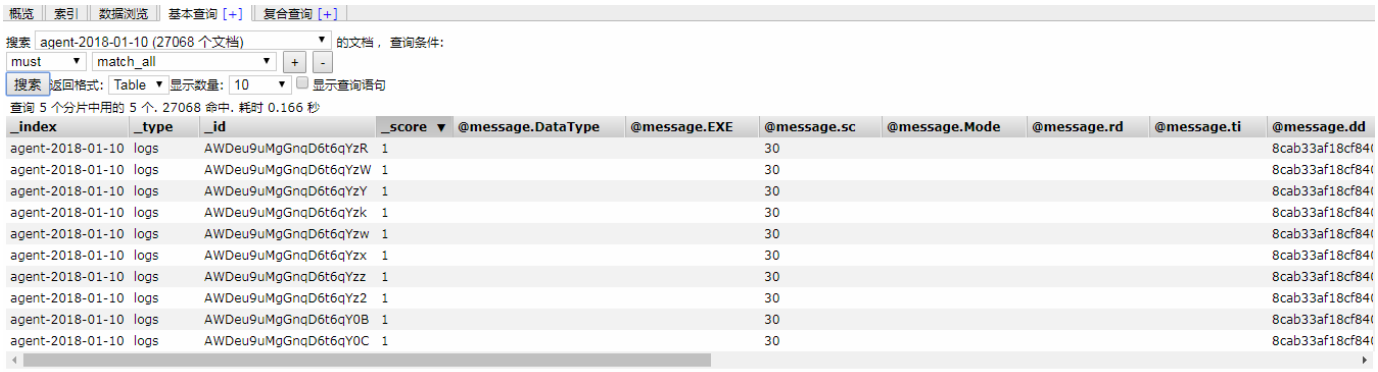
```
F:\dev\apache-flume-1.8.0-bin\apache-flume-1.8.0-bin\bin>flume-ng agent --conf ../conf --conf-file .
./example.conf --name a1

F:\dev\apache-flume-1.8.0-bin\apache-flume-1.8.0-bin\bin>powershell.exe -NoProfile -InputFormat none
 -ExecutionPolicy unrestricted -File F:\dev\apache-flume-1.8.0-bin\apache-flume-1.8.0-bin\bin\flume-
ng.ps1 agent --conf ../conf --conf-file ../example.conf --name a1

Sourcing environment configuration script ../conf\flume-env.ps1
WARN: HADOOP_PREFIX or HADOOP_HOME not found
WARN: HADOOP_PREFIX not set. Unable to include Hadoop's classpath & java.library.path
WARN: HBASE_HOME not found
WARN: HIVE_HOME not found

  Running FLUME agent :
    class: org.apache.flume.node.Application
    arguments: -n a1 -f "F:\dev\apache-flume-1.8.0-bin\apache-flume-1.8.0-bin\example.conf"
```

4. 数据入库到es中



与logstash对比

	Flume	LogStash
		beats

支持数据源	Avro Source Thrift Source Exec Source JMS Source Converter Spooling Directory Source Event Deserializers LINE AVRO BlobDeserializer Taildir Source Twitter 1% firehose Source (experimental) Kafka Source NetCat TCP Source NetCat UDP Source Sequence Generator Source Syslog Sources Syslog TCP Source Multiport Syslog TCP Source Syslog UDP Source HTTP Source JSONHandler BlobHandler Stress Source Legacy Sources Avro Legacy Source Thrift Legacy Source Custom Source	cloudwatch couchdb_changes dead_letter_queue elasticsearch exec file ganglia gelf generator github google_pubsub graphite heartbeat http http_poller imap irc jdbc jms jmx kafka kinesis log4j lumberjack meetup pipe puppet_facter rabbitmq redis relp rss s3 salesforce snmptrap sqlite sqs stdin stomp syslog tcp twitter udp unix varnishlog websocket wmi xmpp
		aggregate alter cidr cipher clone csv date de_dot dissect dns

支持数据处理	Timestamp Interceptor Host Interceptor Static Interceptor Remove Header Interceptor UUID Interceptor Morphline Interceptor Search and Replace Interceptor Regex Filtering Interceptor Regex Extractor Interceptor	drop elapsed elasticsearch environment extractnumbers fingerprint geoip grok i18n jdbc_streaming json json_encode kv metricize metrics mutate prune range ruby sleep split syslog_pri throttle tld translate truncate urldecode useragent uuid xml
支持输出源	HDFS Sink Hive Sink Logger Sink Avro Sink Thrift Sink IRC Sink File Roll Sink Null Sink HBaseSink AsyncHBaseSink	boundary circonus cloudwatch csv datadog datadog_metrics elasticsearch email exec file ganglia gelf google_bigquery graphite graphtastic http influxdb irc juggernaut kafka librato loggly lumberjack metriccatcher mongodb nagios nagios_nsca

	MorphlineSolrSink ElasticSearchSink Kite Dataset Sink Kafka Sink HTTP Sink Custom Sink	opentsdb pagerduty pipe rabbitmq redis redmine riak riemann s3 sns solr_http sqs statsd stdout stomp syslog tcp timber udp webhdfs websocket xmpp zabbix
支持数据缓存	Memory Channel JDBC Channel Kafka Channel File Channel Spillable Memory Channel Pseudo Transaction Channel Custom Channel	