

07

Flume

- Flume简介
- Flume架构
- Flume部署类型



Flume

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。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(比如文本、HDFS、Hbase等)的能力。

- **可靠性**
 - ✓ 保证数据不丢失
- **可扩展性**
 - ✓ 各组件数目可扩展
- **高性能**
 - ✓ 高吞吐量、能满足海量数据收集需求
- **可管理性**
 - ✓ 可动态增加、删除组件
- **文档丰富、社区活跃**
 - ✓ Hadoop生态系统应用广泛



Flume的开源过程

Flume 作为 cloudera 开发的实时日志收集系统，受到了业界的认可与广泛应用。Flume 初始的发行版本目前被统称为 Flume OG (original generation) ，属于 cloudera。但随着 FLume 功能的扩展，Flume OG 代码工程臃肿、核心组件设计不合理、核心配置不标准等缺点暴露出来，尤其是在 Flume OG 的最后一个发行版本 0.94.0 中，日志传输不稳定的现象尤为严重，为了解决这些问题，2011 年 10 月 22 号，cloudera 完成了 Flume-728，对 Flume 进行了里程碑式的改动：重构核心组件、核心配置以及代码架构，重构后的版本统称为 Flume NG (next generation) ；改动的另一原因是将 Flume 纳入 apache 旗下，cloudera Flume 改名为 Apache Flume。

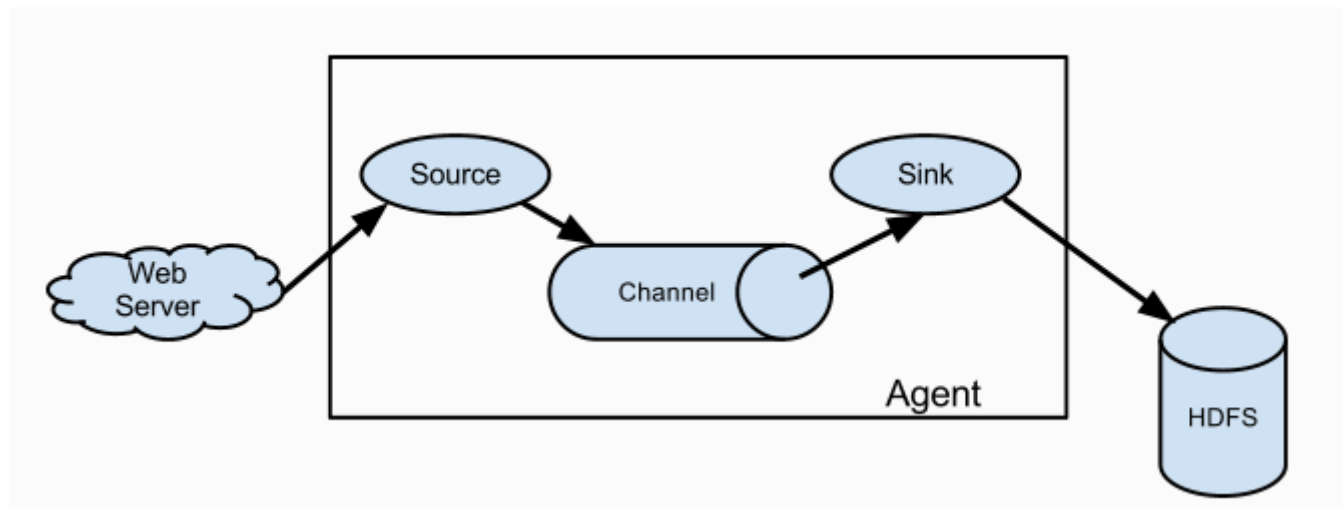
架构

Flume 的核心是把数据从数据源收集过来，再送到目的地。为了保证输送一定成功，在送到目的地之前，会先缓存数据，待数据真正到达目的地后，删除自己缓存的数据。

Flume 传输的数据的基本单位是 Event，Event由Agent外部的Source生成，当Source捕获Event后会进行特定的格式化，然后Source会把Event推入(单个或多个)Channel中。可以把Channel看作是一个缓冲区，它将保存事件直到Sink处理完该事件。Sink负责持久化日志或者把事件推向另一个Source。

架构

Flume 运行的核心是 Agent。Flume以Agent为最小的独立运行单位。一个agent就是一个JVM。它是一个完整的数据收集工具，含有三个核心组件，分别是 Source、Channel、Sink。通过这些组件，Event 可以从一个地方流向另一个地方。



Source

Source 可以接收外部源发送过来的数据。不同的 source，可以接受不同的数据格式。比如有目录池(spooling directory)数据源，可以监控指定文件夹中的新文件变化，如果目录中有文件产生，就会立刻读取其内容。目前Flume支持Avro Source，Thrift Source，Exec Source（即Shell），NetCat Source，Syslog Sources，HTTP Source等大量类型，甚至用户可以自定义Source，使用时仅需在配置文件里声明即可。

Channel

Channel 是一个存储地，接收 Source 的输出，直到有 Sink 消费掉 Channel 中的数据。Channel 中的数据直到进入到下一个 Channel 中或者进入终端才会被删除。当 Sink 写入失败后，可以自动重启，不会造成数据丢失，因此很可靠。

可供选择的 Channel 有 Memory, JDBC, File。

- Memory 可以实现高速的吞吐，但是无法保证数据的完整性。
- FileChannel 保证数据的完整性与一致性。在具体配置 FileChannel 时，建议 FileChannel 设置的目录和程序日志文件保存的目录设成不同的磁盘，以便提高效率。

Sink

Sink负责从Channel中取出并发送数据给外部源或者其它Source。如数据可以写入到 HDFS 或者 HBase 中。Flume当前支持HDFS、Logger、Avro、Thrift, IRC, File, HBase等大量类型 Sink。

Sink 在设置存储数据时，可以向文件系统、数据库、hadoop 存数据，在日志数据较少时，可以将数据存储存储在文件中，并且设定一定的时间间隔保存数据。在日志数据较多时，可以将相应的日志数据存储到 Hadoop 中，便于日后进行相应的数据分析。

可靠性

当节点出现故障时，日志能够被传送到其他节点上而不会丢失。Flume提供了三种级别的可靠性保障，从强到弱依次分别为：

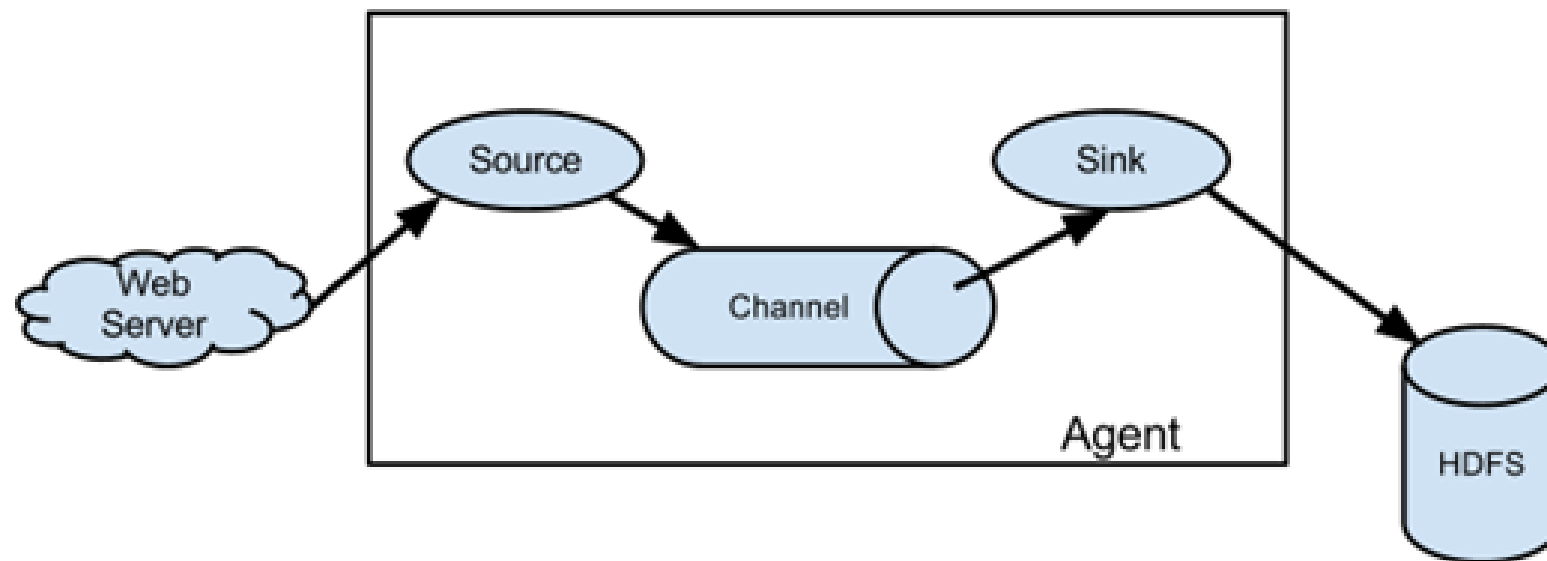
- end-to-end（收到数据agent首先将event写到磁盘上，当数据传送成功后，再删除；如果数据发送失败，可以重新发送。）
- Store on failure（这也是scribe采用的策略，当数据接收方crash时，将数据写到本地，待恢复后，继续发送）
- Best effort（数据发送到接收方后，不会进行确认）。

可恢复性

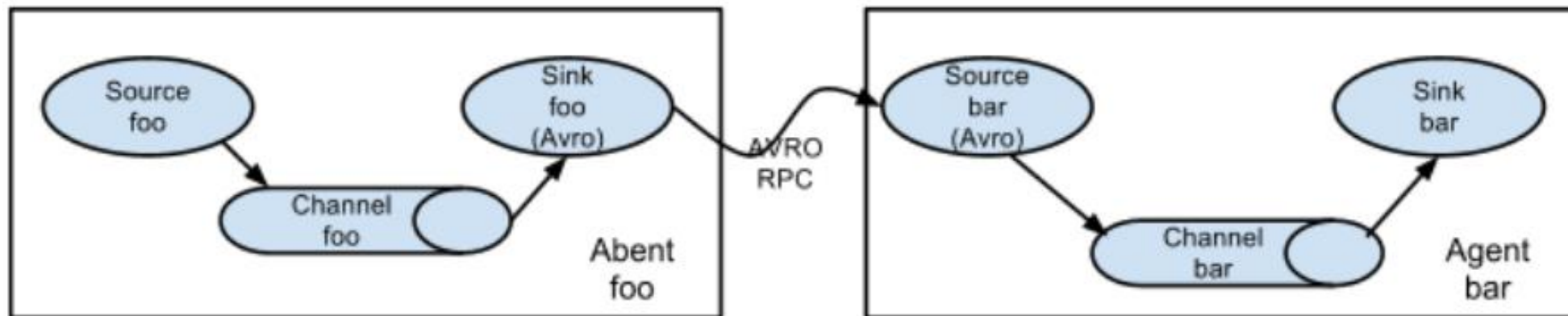
还是靠Channel。推荐使用FileChannel，事件持久化在本地文件系统里(性能较差)。

Flume的部署类型

单一流程



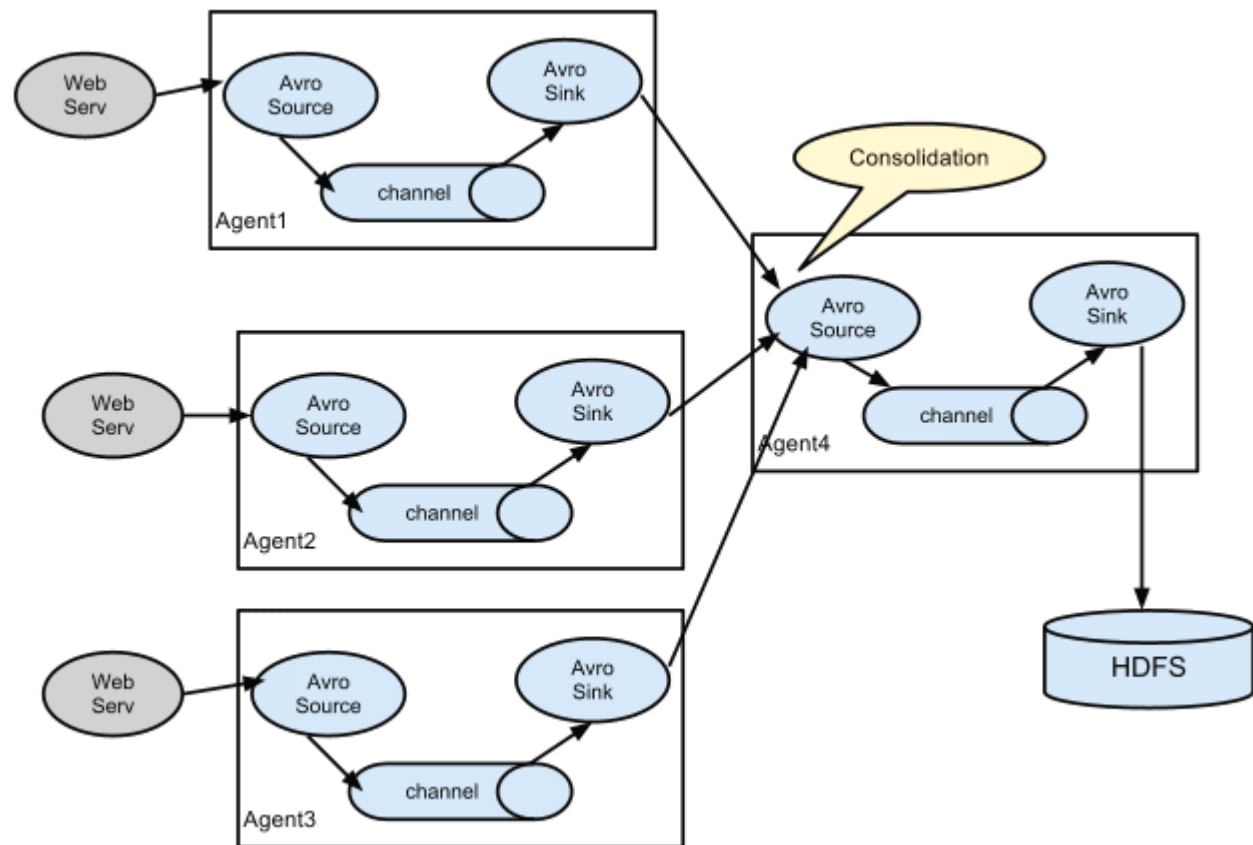
多Agent流程



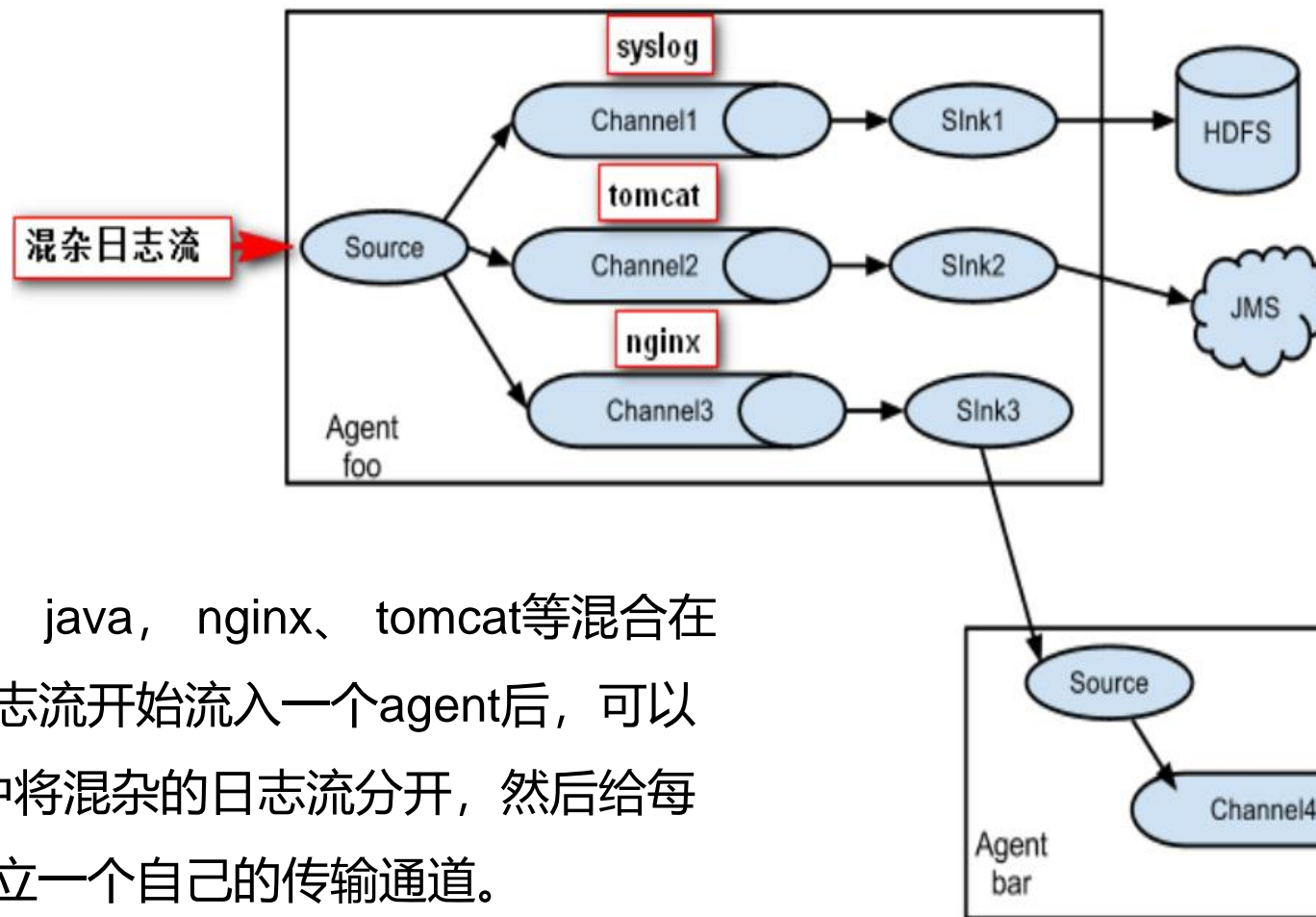
可以将多个Agent顺序连接起来，将最初的数据源经过收集，存储到最终的存储系统中。这是最简单的情况，一般情况下，应该控制这种顺序连接的Agent 的数量，因为数据流经的路径变长了，出现故障将影响整个Flow上的Agent收集服务。

流的合并（多个Agent的数据汇聚到同一个Agent）

这种情况应用的场景比较多，比如要收集Web网站的用户行为日志，Web网站为了可用性使用的负载集群模式，每个节点都产生用户行为日志，可以为每个节点都配置一个Agent来单独收集日志数据，然后多个Agent将数据最终汇聚到一个用来存储数据存储系统，如HDFS上。

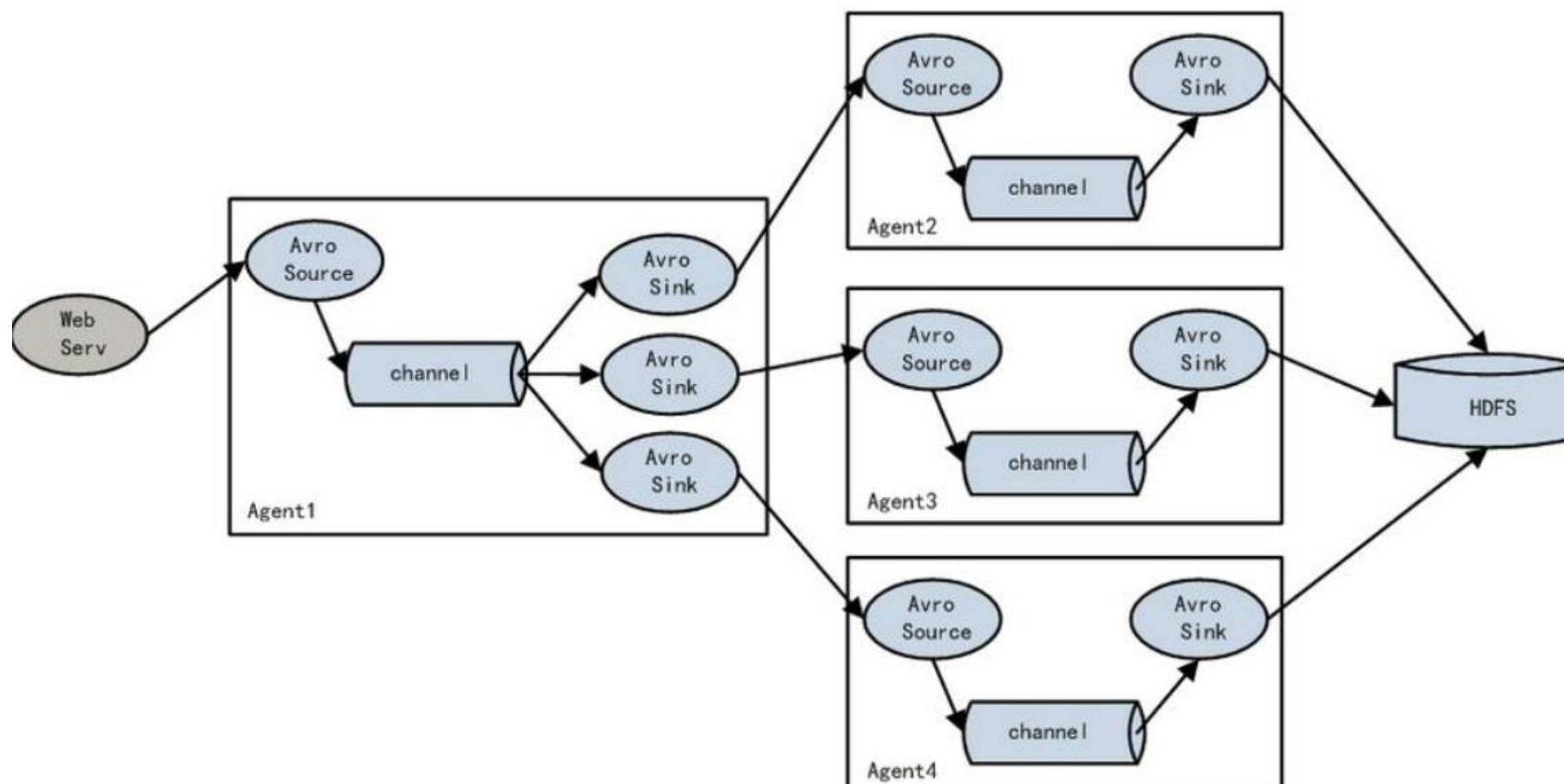


多路复用流（多级流）



当syslog, java, nginx、tomcat等混合在一起的日志流开始流入一个agent后，可以在agent中将混杂的日志流分开，然后给每种日志建立一个自己的传输通道。

负载均衡



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。