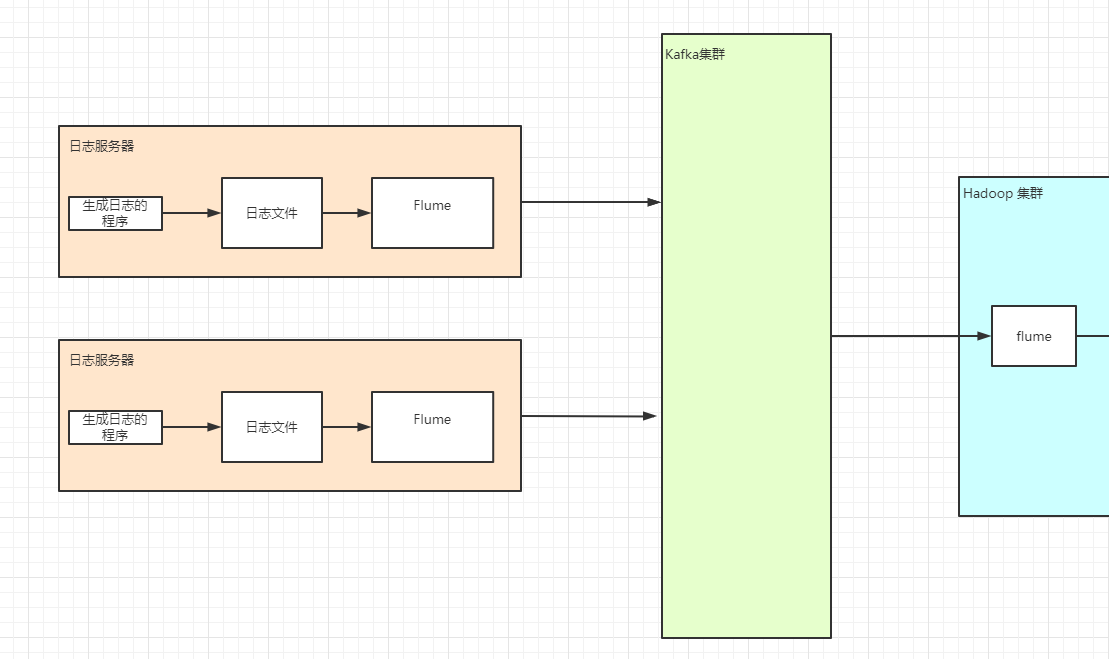
因为我们的数据是模拟的埋点日志，为了更好的模拟真实开发环境，采用的采集架构为 flume – kafka – flume – hdfs，没有直接采用 fume-hdfs 的原因是在真实环境中，日志可能来源于多个服务器，如果直接用 flume 传输到 hdfs，那么会因为日志的输入量及速度过大，导致接收服务器承受不了这么高的压力而丢失数据，kafka作为一种高可靠的消息队列，可以很好的解决这个问题。



（架构图）

为了配置合适的flume配置，我们上官查询了相关资料，对 flume 的各种常用 source、channel做了对比（因为 sink 一定是选用 hdfsSink，所以就在这里没有介绍）：

|  |  |  |
| --- | --- | --- |
| Source 类型 | 介绍 | 使用场景 |
| Avro Source | 监听Avro端口并从外部Avro客户端流接收事件。 与另一个（以前的跃点）Flume代理上的内置Avro Sink配对时，它可以创建分层的集合拓扑。 | RPC异构流数据交换 |
| Thrift Source | 监听Thrift端口并接收来自外部Thrift客户端流的事件。 与另一个（以前的跃点）Flume代理上的内置Thrift Sink配对时，它可以创建分层的集合拓扑。 可通过启用kerberos身份验证将节俭源配置为以安全模式启动。 agent-principal和agent-keytab是Thrift源用来验证kerberos KDC的属性。 | RPC异构流数据交换 |
| Exec Source | Exec source在启动时运行给定的Unix命令，并期望该进程在标准输出上持续生成数据(stderr被丢弃，除非属性logStdErr被设置为true)。如果进程出于任何原因退出，源也会退出，并且不会产生进一步的数据。这意味着cat [named pipe]或tail - f [file]这样的配置将生成所需的结果，而as date可能不会——前两个命令生成数据流，而后者生成单个事件并退出。 | 监听文件，不支持断点续传 |
| Spooling Directory Source | Spooling Directory Source通过监听某个目录下的新增文件，并将文件的内容读取出来，实现日志信息的收集。 | 监听文件夹中产生的新文件，支持断点续传 |
| TailDir Source | 监视指定的文件，一旦检测到附加到每个文件的新行，就几乎实时跟踪它们。如果正在写入新行，改 Source 将重新尝试读取它们，等待写入完成。 | 实时监听文件，支持断点续传 |
| Kafka Source | Kafka Source是一个Apache Kafka消费者，它读取来自Kafka主题的消息。如果有多个Kafka Source 在运行，可以将它们配置为相同的使用者组，这样每个使用者都将为主题读取一组惟一的分区。 | Flume 作为 Kafka 消费 |
| NetCat TCP Source | 监听指定服务器:端口产生的数据。 | 监听网络端口 |

|  |  |  |
| --- | --- | --- |
| Channel 类型 | 简介 | 使用场景 |
| Memory Channel | 事件被写入到内存队列中 | 一般用于 Source -Channel-Sink这种结构，并要求延时低 |
| File Channel | 事件被写入到文件中 | 一般用于 Source-Channel-Sink这种结构，对时间要求不高，对数据的安全性要求 高 |
| JDBC Channel | Flume 事件被存储在数据库里 | Source - Channel 这种结构，Flume 到数据库 |
| Kafka Channel | lume 事件被存储到 Kafka 集群中，它与Flume的 Source和 Sink结合有三种方式：① Flume Source -> Kafka Channel ->Flume Sink，此时 Kafka 充当一个高可用的传输通道 ② Flume -> Kafka Channel，直接把数据写到 Kafka 集群中，其他应用程序可以作为消费者来消费这个数据(用这种方式不用配置 Flume Sink) ③ Kafka -> Flume Sink，就是 Source 可以不用来自 Flume 的Source，只要 写到 Kafka 的 topic 中就行，这种方式提供了一种低延时、高容错的方式把 event 从 Kafka 传到 Flume Sinks，比如 HDFS、HBase 或 Solr。 |  |

因为我们模拟埋点日志是往文件中写入信息，所以要从 Exec Source 和 TailDir Source 中选一个，而 TailDir Source 支持**断点续传**，所以就选择 **TailDir Source**；我们在传输日志的时候是把 Flume 中的文件写到 Kafka 中，所以选择 Kafka Channel ，然后在下一个 Flume 的 Source 中使用 Kafka Source 来读取 Kafka topic 里的数据。

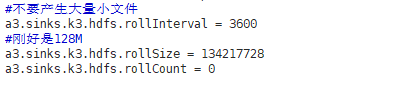
Flume 的配置：在 flume-sink 中，**设置输出按照日期滚动：**



设置压缩格式为 lzo 压缩，因为经过查询，**lzo 的压缩率最高**：



为了避免在hdfs中生成大量小文件，设置**生成一个文件的阈值：**



为了提高 kafka 的并行读及可靠性，设置kafka的分区为2，并行度为 3

