108

写入到 Hadoop 1 和 Hadoop 2*

Flume 可以写入到 Hadoop 1 或 Hadoop 2 的 HDFS。Apache 软件基金会发布的二进制工件是针对 Hadoop 1 的。如果数据被写入到 Hadoop 2.x 集群，那么用户必须使用命令 `mvn clean install -Dhadoop.profile=2` 重新编译 Flume（该命令要求基于 Hadoop 2.x 编译的 HBase 0.94.2 在 Maven 缓存中是可用的），或使用命令 `mvn clean install -Dhadoop.profile=hbase-98`，用 Hadoop 2.4.0 和 HBase 0.98 编译。Flume 与 Flume 的二进制文件中没有打包 Hadoop 及其依赖关系。如果 Hadoop 已经安装在机器上，Flume 会自动获得依赖；否则，用户必须在 Flume 的环境变量中添加包含 HDFS 客户端库及其依赖关系的目录。Hadoop 供应商通常发布 HDFS 客户端库的版本号，它们会使用该版本号发布 Flume 二进制文件。

使用序列化器控制数据格式 *

写出到 HDFS 的数据最终会被各种各样的其他系统所使用。因此，HDFS Sink 足够灵活去支持可以被这些系统理解的数据格式是很重要的。HDFS Sink 通过允许用户嵌入序列化器来转换 Flume 事件为处理事件的系统可以理解的格式，并将它们写出到最终会刷新到 HDFS 的流中，这样就能允许用户以适用的的数据格式写入 HDFS。Flume 自带了一些序列化器，支持常见的格式 text 和 Avro 等。需要记住的是，只有当 HDFS Sink 配置好了事件使用数据流或压缩流，序列化器才启用。本节将说明如何编写一个序列化器，以及如何告诉 HDFS Sink 使用序列化器。

文件格式

当写数据到HDFS，或数据可能被应用处理且并不意味着人类可读的任何其他系统，使用例如 Avro、Protobuf 等二进制格式更有意义。就磁盘占用的空间量和写出数据花费的时间量而论，二进制格式往往更有效率。这是因为二进制格式可以更有效地编码数据，在 Avro 中一个整数四个字节，而如果使用 text，如果数量超过四位数，它将超过四个字节（在普通的旧的 ASCII 中）。多种二进制格式在写之前还能压缩数据。

Flume 的二进制格式还有另一个特别大的优势。Flume 批量写出事务到 HDFS，在
109 需要时分配块。如果由于其他任何原因一个块分配失败或者写失败，可能会写入部分事件。当这样的失败发生，Flume 将在一个新文件重试该事件，但部分事件仍将在HDFS 文件里。如果这部分事件代表Hive 消费的一行数据，Hive 查询可能会失败，因为数据可能没有意义，也可能是不完整的。甚至更糟糕的是，如果解析没有失