MapReduce 作业可以直接使用 Avro MapReduce 模块读取 Avro container 文件。Hive、Pig、Cloudera Impala 等其他系统也支持 Avro container 格式。为了压缩 Avro container 文件，应该使用 Avro 的本地压缩而不是使用 Flume 的压缩流。为了使用 Avro 的本地压缩，将 `compressionCodec` 参数的值设置为对应的压缩编码 `deflate` 或 `snappy`，再将该参数传递给序列化器。Avro container 格式允许用户指定同步标志之间的数据总量，可以使用 `syncIntervalBytes` 设置，该参数指定了以字节为单位。默认大小是 `2048000`（2 MB）。下面展示了一个 Avro 序列化器配置的例子：


```
agent.sinks.hdfsSink.fileType = DataStream
agent.sinks.hdfsSink.serializer = AVRO
agent.sinks.hdfsSink.serializer.syncIntervalBytes = 4096000
agent.sinks.hdfsSink.serializer.compressionCodec = snappy
```

`EventSerializer` 接口是 `flume-ng-core` 工件的一部分，可以像例 3-6 所示的那样，添加到你的序列化器的 *pom.xml* 文件中。

Ingest 格式和 Output 格式的不同 *

当使用 Flume 时混乱的主要来源是格式之间的关系，包括写入到最终目的地的数据格式和接收的数据格式。**接收格式**可以是任何格式——取决于使用的 Source，输入数据转化为 Flume 事件的方式会有所不同。例如，如果通过 RPC 客户端和 RPC Source 接收数据，应用程序将数据转换成 Flume 事件。对于其他 Source，如 spooling directory，一个可插件化的组件完成从原始数据格式到 Flume 事件的转换。

一旦事件在 Flume 里，对于 Flume 事件数据就像一个黑匣子，直到它到达目的 Sink。这条规则的一个异常是什么时候用**拦截器**，拦截器是实际上可以修改事件的组件。我们将在第 6 章讨论拦截器。在写数据到存储系统时，Flume 事件本身需要转换成处理系统使用的格式，处理系统用于从存储系统中读取数据。这是**输出格式**。大多数扮演终端 Sink 作用的 Sink，例如 HDFS、HBase 和 Morphline Solr Sink，都接受可以将 Flume 事件转换为最终目的地格式的插件。

正如你所看到的，在 Source 中有一个从原始格式到 Flume 事件的初始转换，第二个转换成最终目的地格式发生在目的地 Sink 中。如果最初和最终的格式都是相同的，那么只是将原来的格式转换为一个字节数组，然后以预编码的数据写字节数组，这种方式是有意义的。