就会被滚动（例如，即使在保存间隔达到之前事件的数量达到了计数的数量，那么文件会被滚动）。所以，同时可以启用不止一个参数。

当使用基于时间的分桶，有可能在某个固定的时间点之后，就不会有事件再写入到bucket中。如果对于文件关闭启用了滚动间隔，它将至少需要滚动间隔的时间。如果滚动间隔没有启用，这样的文件可能从不会被关闭。所以，总是推荐设置`hdfs.idleTimeout`，它表示在最后一个事件写入文件之后关闭文件要等待的秒数值时间。大多数情况下，将这个值设置比滚动间隔小得多更有意义，所以当没有数据再写入到bucket时，比关闭间隔更短的时间的数据就是可用的。但是它应该设置得比事件写入bucket的平均时间稍微大一些，这样文件不会被要求更频繁地关闭文件——不然，HDFS上会有很多小文件，给HDFS NameNode造成压力。

当有多个HDFS Sink写入HDFS，在同一个Agent或不同的Agent上，每个Agent写入到不同的目录或使用不同的文件前缀是很重要的。重要的原因是确切地说一个HDFS Sink只能写入一个文件；其他试图写入相同文件的HDFS Sink可能会遇到异常且不能再写数据。为了避免这个问题，每个HDFS Sink拥有它正在写入的目录且没有其他写入的Sink或进程，建议使用这种方式分桶数据。这可以通过使用主机名和Sink名作为bucket名字的一部分来完成（主机名可以通过Flume自带的主机拦截器来插入），或者使用多路复用Channel选择器来确保每个HDFS Sink写的数据属于一个确定的主题等。 105

HDFS Sink允许用户以顺序文件或压缩文件，或者任何二进制或文本的格式写数据。`hdfs.fileType`参数控制了文件格式。如果要以顺序文件写数据，就将该值设置为`SequenceFile`。对于顺序文件，Flume写事件主体作为数值键的对应值，写为`LongWritable`类型。如果事件报头中有时间戳，可以使用时间戳作为键；否则，就是用当前事件的毫秒值。事件主体基于`hdfs.writeFormat`参数将它本身写出为`ByteWritable`或`TextWritable`类型。如果要使用`textWritable`，该参数应该被设置为`text`；对于`ByteWritable`则应设置为`writable`。当使用顺序文件，`serializer`和`serializer.*`参数将被忽略。当使用数据流或压缩流时，`hdfs.writeFormat`参数会被忽略。

HDFS Sink允许用户指定序列化器来转换事件为用户适用的数据格式。只有当文件类型设置为`DataStream`或`CompressedStream`，序列化器才会启用。然后序列化器可以以它选择的格式将数据写到磁盘。序列化器内部可以做任何压缩，但是Flume本身适用数据流的时候不压缩写入的数据。为了压缩数据，文件类型必须设置为CompressedStream，`hdfs.codeC`参数必须设置为`gzip`、`bzip2`、`lzo`、`lzop`或`snappy`中的一种，指定了写入到HDFS使用的压缩编码。当使用数据流时压缩编码不需要设置。当使用压缩流时，如果文件后缀没有指定，对于配置好压缩编码的写入文件将使用恰当的扩展名（文件后缀