85 置这个参数。它抑制写入或读取速率的安全值。当读取速率远大于写入速率，节流速率是很有用的，反之亦然。

下面的配置展示了一个配置能保存多达 100000 个事件的 Memory Channel，每个事务能保存 1000 个事件。Channel 中所有事件占有的内存可以是大约 5GB 空间的最大值。在这 5GB 当中，Channel 将 10% 为事件 header 保留（正如 byteCapacityBufferPercentage 参数所指明的），使得事件主体可以利用 4.5GB 大小。

```
agent.channels = mc
agent.sources = sq

agent.channels.mc.type = memory
agent.channels.mc.capacity = 100000
agent.channels.mc.transactionCapacity = 1000
agent.channels.mc.byteCapacity = 5000000000
agent.channels.mc.byteCapacityBufferPercentage = 10

agent.sources.sq.type = seq
agent.sources.sq.channels = mc
```

File Channel

File Channel 是 Flume 的持久 Channel。它将所有事件写到磁盘，因此在程序关闭或机器宕机的情况下不会丢失数据。File Channel 保证了即使机器或 Agent 宕机或重启，只有当 Sink 取走了事件并提交给事务时，任何提交到 Channel 的事件才从 Channel 移除。它被设计为高并发且可以同时处理多个 Source 和 Sink。File Channel 的设计大致上是基于 Rosenblum 和 Ousterhout 的关于日志结构文件系统的论文 [lfs-paper]。更多关于该设计的细节将在本节稍后讨论。

File Channel 被设计用于数据需要持久化和不容忍数据丢失的场景下。因为 Channel 将数据写到磁盘，它不会由于宕机或失败造成数据丢失。一个额外的好处，因为它写数据到磁盘，Channel 可以有非常大的容量，尤其是和 Memory Channel 相比。

只要磁盘空间可用，File Channel 可以有非常大的容量，能保存多达几十或几百个成千上万的事件。当预计的 Sink 从 Channel 取走事件的速率无法跟上一个有限的高峰期时，File Channel 对可能存在的大量待处理事件是特别有用的。因为一旦事件被提交，Channel 就不在内存中保存它们了，File Channel 比相等容量的 Memory Channel 需要更少的堆栈空间。

86 File Channel 保证每个写入的事件，经过 Agent 和机器故障或重新启动，仍然是可用的。它通过将每个放入 Channel 的事件写出到磁盘上。一旦提交一个事务，事务中的事件对