

Channel

Channel 是位于 Source 和 Sink 之间的缓冲区。因此，Channel 允许 Source 和 Sink 运作在不同的速率上。Channel 是 Flume 保证数据不丢失的关键（当然，这是在正确配置的情况下）。Source 写入数据到一个或多个 Channel 中，再由一个或多个 Sink 读取。Sink 只能从一个 Channel 读取数据，而多个 Sink 可以从相同的 Channel 读取以获得更好的性能。Channel 有事务语义，允许 Flume 为写进 Channel 的数据提供明确的保证。

位于 Source 和 Sink 之间作为缓冲区的 Channel 操作，有几个优点。Channel 允许 Source 在同一 Channel 上操作，以拥有自己的线程模型而不必担心 Sink 从 Channel 读取数据，反之亦然。位于 Source 和 Sink 之间的缓冲区也允许它们工作在不同的速率，因为写操作发生在缓冲区的尾部，读取发生在缓冲区的头部。这也使得 Flume Agent 能处理 Source “高峰小时”的负载，即使 Sink 无法立即读取 Channel。

Channel 允许多个 Source 和 Sink 在它们上面进行操作。Channel 本质上是事务性的。每次从 Channel 中写入和读取数据，都在事务的上下文中发生。只有当写事务被提交，事务中的事件才可以被任意 Sink 读取。同样，如果一个 Sink 已经成功读取了一个事件，该事件对于其他 Sink 是不可用的，除非该 Sink 回滚事务。

在大多数实时应用程序临时负载出现峰值是很常见的，Flume 被设计用来处理这些情况。事件将缓冲到 Channel 直到 Sink 移除它们，让 Agent 处理传入负载的变化。每个 Agent 可以处理多少额外的数据取决于 Channel 的容量。分配 Channel 的容量应根据预期的所有写入 Channel 的 Source 最大峰值负载和所有 Sink 的读取速度的总和。这个设计还允许 Source 和 Sink 对失败具备重试逻辑。对于失败，Source 可以尝试再次写入 Channel，Sink 可以尝试再次读取。