

写入和读取操作将按顺序应用到队列。

所有代表现在不完整事务称为 *inflight* 的指针也被写入到磁盘，所以检查点时刻的任何被读取但还没提交的事件，都可以在加载检查点后重新插入到队列。检查点时刻仍然运行中的写入事务也可以通过使用 *inflight* 文件进行恢复（可能有检查点之前但是提交之后的数据文件，它们在检查点时刻没有被插入到队列，因此没有从数据文件回放，因为数据文件只能在检查点时刻的偏移量中回放）。

92

一旦回放完成，Channel 就为正常操作做好了准备。当没有队列存在或当它是不完整时，Channel 就进行全回放。前面讨论过，Channel 可以在检查点完成之后立即备份，所以如果现在的检查点损坏，或者 Agent 在完成之前被杀死，那么可以加载备份检查点来避免全回放。

总结

本章我们讨论的 *Channel*，是位于 Source 和 Sink 之间的缓冲区，保存通过 Source 传入到 Flume Agent 的数据，直到它被 Sink 移除。Channel 可以在内存或磁盘，使用内存的 *Memory Channel* 拥有更好的性能，同时基于磁盘的 *File Channel* 保证了持久化，Agent 和机器通过持久化所有操作和数据到磁盘进行重启。

在下一章，我们将讨论 Sink 是如何设计的，以及 Flume 自带的不同种类的 Sink。我们也将讨论如何实现自定义 Sink。

参考文献

- [lfs-paper] “The design and implementation of a log-structured file system,” <http://bit.ly/1wxCsoM>