

务表明 Channel 可以删除在事务的上下文中读取的事件。当 Agent 重启时, Sink 中缓冲的这些事件可能会造成数据丢失, 因为数据没有在持久的存储或缓冲中, 而是在可能不能重用的 Sink 的内部数据结构中。一旦提交, Sink 必须保证事件已经被提交到下一个 Agent 的 Channel (在 RPC Sink 的情况下) 或已经持久化到最终目的地。

优化 Sink 的性能

通常一个线程运行一个 Sink : Sink 运行器。因为一个线程运行一个 Sink, 且大多数 Sink 倾向于 I/O 密集, 可能会有这样的情况产生, 当 Sink 等待 I/O 完成且 Channel 中没有将要移除的事件。因此, 在大多数情况下, 当个别 Sink 比写入到指定 Channel 的速率要慢很多时, 那就值得添加多个 Sink 用来写入到相同目的地和从相同 Channel 读取事件。这对于 HDFS 和 HBase 的 Sink 尤其有用, 能用来提升写入到 HDFS 和 HBase 的速率。

性能优化所需的 Sink 数量取决于不同的因素, 包括正被使用的 Sink、目的地 (Avro Source/HDFS/HBase)、网络吞吐量、Channel、Channel 使用的磁盘 I/O 性能。因此, 在确定数量之前尝试各种不同数量的 Sink 配置通常是一个好主意。当增加 Sink 数量来提升性能时, 必须确保的一点是资源不会因过度使用而结束, 创建有太多上下文切换或网络的情况下将会导致阻塞。当达到这一点, 说明了机器上的硬件被最大化利用; 如果 HDFS 或 HBase 集群仍然有能力, 可以在其他机器上添加更多的 Flume Agent。

写入到 HDFS : HDFS Sink

96

Flume 主要被设计为以可扩展的方式写入数据到 HDFS。通常用户有成百上千的数据源写数据到 HDFS。如果所有的 Source 直接将数据写入 HDFS, 那么通常会给 NameNode 和 HDFS 集群造成很大压力。因此建议使用 Flume Agent 的多个层写入数据。这在第 2 章的“复杂流”一节已经解释过。Agent 将使用 HDFS Sink 最终将数据写入到 HDFS。HDFS Sink 是很灵活的, 可以基于事件的报头、事件的时间戳 (或 Sink 的当前时间) 等, 配置写入到不同的目录。我们将在稍后一节中看到 Sink 关于批量事件到不同目录的工作细节。

HDFS Sink 允许用户通过编写和部署的序列化器, 自定义写入到 HDFS 事件的格式。HDFS Sink 支持 Hadoop 1 和 Hadoop 2, 但是代码必须针对 Hadoop 的相关版本进行特定的编译。例如, Flume 针对 Hadoop 2 进行的编译将不能将数据写入到 Hadoop 1 HDFS 集群。这是因为在 Hadoop 1 和 Hadoop 2 之间二进制不兼容性变化。因此, 针对 HDFS 特定版本编译的二进制文件必须在特定的 HDFS 中使用。Apache Flume 目前只发布针对 Hadoop 1 编译的二进制, 但是有一些供应商, 基于 Hadoop 2 的发行版, 发布可编制的