第2章 <7>

用Apache Flume处理流数据

用中间系统推送数据到 HDFS 和类似的存储系统是非常普遍的使用案例。例如 Apache Flume、Apache Kafka、Facebook 的 Scribe 等几个系统，都支持这种使用案例。这些系统允许 HDFS 和 HBase 集群处理数据偶然突发的情况而没必要具备处理持续写入速度的能力。这些系统在数据生产者和最终目的地之间担当缓冲的角色。凭借其缓冲能力，它们能够平衡在生产者和消费者之间的阻抗不匹配，因此可以提供一个稳定的流状态。扩展这些系统比扩展 HDFS 或者 HBase 集群更容易。这样的系统也允许应用程序推送数据，而不用担心万一 HDFS 放降，必须缓冲数据和重试等。

大多数这种系统都有一些基本类似之处。通常，这些系统通过 RPC 调用成 HTTP（可以通过客户端 API 展现），接收生产者数据的组件，通过 RPC 调用或 HTTP（可以通过客户端 API 展现）。它们也有作为缓冲区的组件，数据存储在缓冲区直到数据被移动到下一阶段或目的地。在这一章中，我们将会讨论 Flume Agent 的基本架构和如何配置 Flume Agent 来将数据从各种各样的应用中移动到 HDFS 或者 HBase。

在大企业中，Apache Hadoop 正在变成一个标准的数据处理框架。应用程序经常产生大规模数量级的数据，并写入到 HDFS——Hadoop 基础的分布式文件系统。Apache Flume 被构思为这样一个系统，以一种可靠的和可扩展的形式将数据写入 Apache Hadoop 和 Apache HBase。因此，Flume 的 HDFS Sink 和 HBase Sink 提供了一个非常丰富的功能集合，使得它可以用这些系统支持的任意格式和 MapReduce/Hive/Impala/Pig 友好型的方式写入数据。本书将论述我们为什么需要一个类似 Flume 的系统、Flume 的设计和实现，以及使得 Flume 高扩展、灵活和可靠的各种特征。 <8>

我们需要 Flume

为什么我们真的需要一个类似 Flume 的系统呢？为什么不是简单地从每个生产数据的应