Storm的特征

Storm是一个开源的分布式实时计算系统，可以简单、可靠地处理大量的数据流。Storm支持水平扩展，具有高容错性，保证每个消息都会得到处理，而且处理速度很快（在一个小集群中，每个节点每秒可以处理数以百万计的消息）。Storm的部署和运维都很便捷，而且更为重要的是，可以使用任意编程语言来开发应用。

1. 编程模型简单

在大数据处理方面，Hadoop为开发者提供了MapReduce原语，使并行批处理程序变得非常简单和优美。同样，Storm也为大数据的实时计算提供了一些简单优美的原语，这大大降低了开发并行实时处理任务的复杂性，可以快速、高效地开发应用。

1. 可扩展

在Storm集群中真正运行Topology的主要有3个实体：工作进程、线程和任务。

Storm集群中的每台机器都可以运行多个工作进程，每个工作进程又可创建多个线程，每个线程可以执行多个任务，任务是真正进行数据处理的实体，开发的Spout、Bolt就是作为一个或者多个任务的方式执行的。因此，计算任务在多个线程、进程和服务器之间并行进行，支持灵活的水平扩展。

1. 高可靠性

Storm可以保证Spout发出的每条消息都能被“完全处理”，这也是直接区别于其他实时系统的地方，如S4。

注意：Spout发出的消息后续可能会触发产生成千上万条消息，可以形象地理解为一棵消息树，其中Spout发出的消息为树根，Storm会跟踪这棵消息树的处理情况，只有当这棵消息树中的所有消息都被处理了，Storm才会认为Spout发出的这个消息已经被“完全处理”。如果这棵消息树中的任何一个消息处理失败了，或者整棵消息树在限定的时间内没有“完全处理”，那么Spout发出的消息就会重发。

考虑到尽可能减少内存的消耗，Storm并不会跟踪消息树中的每个消息，而是采用了一些特殊的策略，它把消息树当作一个整体来跟踪，对消息树中所有消息的唯一ID进行异或计算，通过是否为0来判定Spout发出的消息是否被“完全处理”，这极大地节约了内存并简化了判定逻辑，后面会详细介绍这种机制。

在这种模式下，每发送一个消息，都会同步发送一个ack/fail，对于网络的带宽会有一定的消耗，如果对可靠性要求不高，则可使用不同的emit接口关闭该模式。

上面所说的，Storm保证了每个消息至少被处理一次，但是对于有些计算场合，会严格要求每个消息只被处理一次，Storm的0.7.0引入了事务性拓扑，解决了这个问题，后面章节会详述。

1. 高容错性

如果在消息处理过程中出了一些异常，Storm会重新安排这个出问题的处理单元。Storm保证一个处理单元永远运行（除非显式杀掉该处理单元）。当然，如果处理单元中存储了中间状态，那么当处理单元重新被Storm启动时，需要将自身处理的中间状态恢复。

1. 支持多种编程语言

除了用Java实现Spout和Bolt，还可以使用其他编程语言来完成这项工作，这一切得益于Storm的多语言协议。多语言协议是Storm内部的一种特殊协议，允许Spout或Bolt使用标准输入和标准输出来传递消息，传递的消息为单行文本或多行JSON编码的格式。

Storm支持多语言编程主要是通过ShellBolt、ShellSpout和ShellProcess这些类来实现的，这些类都实现了IBolt和ISpout接口，以及让Shell通过Java的ProcessBuilder类来执行脚本或者程序的协议。

可以看到，采用这种方式，每个Tuple在处理时都需要进行JSON的编解码，因此在吞吐量上会有较大影响。

1. 支持本地模式

Storm有一种“本地模式”，也就是在进程中模拟一个Storm集群的所有功能，以本地模式运行Topology与在集群上运行Topology类似，这对于开发和测试来说非常有用。

1. 高效

永ZeroMQ作为底层消息队列，保证消息能被快速处理。