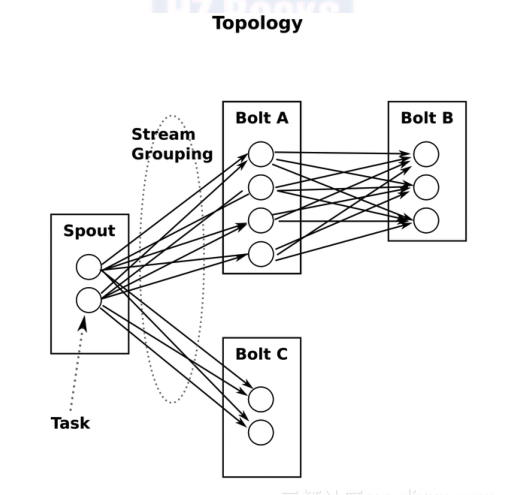
Topology拓扑

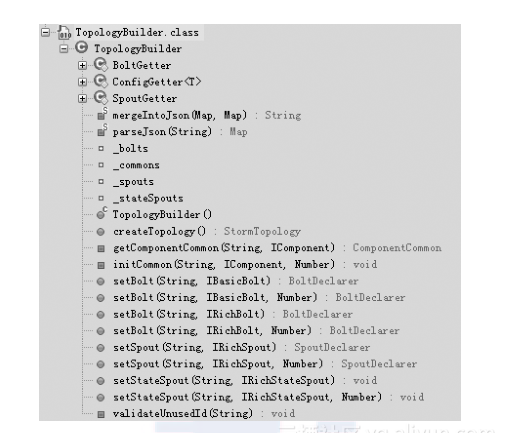
Hadoop上运行的是MapReduce作业，而在Storm上运行的是拓扑Topology，这两者之间是非常不同的。一个关键的区别是：一个MapReduce作业最终会结束，而一个Topology拓扑会永远运行（除非手动杀掉）。

从字面上解释Topology，就是网络拓扑，是指用传输介质互连各种设备的物理布局，是构成网络的成员间特定的物理的（即真实的），或者逻辑的（即虚拟的）排列方式。拓扑是一种不考虑物体的大小、形状等物理属性，而只使用点或者线描述多个物体实际位置与关系的抽象表示方法。拓扑不关心事物的细节，也不在乎相互的比例关系，只是以图的形式表示一定范围内多个物体之间的相互关系。从Storm角度考虑，它不是网络拓扑，但是又类似于网络拓扑的结构，所以取名Topology。

那么Storm的Topology指的是类似于网络拓扑图的一种虚拟结构。Storm的拓扑Topology类似于MapReduce任务，一个关键的区别是MapReduce任务运行一段时间后最终会完成，而Storm拓扑一直运行（直到杀掉它）。一个拓扑是由Spout和Bolt组成的图，Spout和Bolt之间通过流分组连接起来。下图形象地描述了Topology中的Spout和Bolt之间的关系。



通过对上图的理解可以看出，Topology是由Spout、Bolt、数据载体Tuple等构成的一定规则的网络拓扑图。Storm提供了TopologyBuilder类来创建Topology。打个比方，TopologyBuilder是Topology的骨架，Spout、Bolt是Topology的肉和血液。TopologyBuilder类的主要方法如下图所示。



TopologyBuilder实际上是封装了Topology的Thrift接口，也就是说Topology实际上是通过Thrift定义的一个结构，TopologyBuilder将这个对象建立起来，然后Nimbus实际上运行一个Thrift服务器，用于接收用户提交的结构。由于采用Thrift实现，所以用户可以用其他语言建立Topology，这样旧提供了比较方便的多语言操作支持。

Topology实例

下面从一个简单的例子开始介绍Topology的构建和定义，通过此案例能够基本理解Storm，并且能够构建一个简单的Topology。本实例使用Topology来统计一个句子中单词出现的频率。下面详细介绍如何设计和运行Topology，以及一些注意事项。

1. 设计Topology结构
2. 设计数据流
3. 代码实现
4. Topology运行

Topology运行有两种模式：本地模式和分布式模式。这两种模式的接口区别很大，使用场景也不相同。另外，下面还将介绍Topology的运行流程、方法调用过程以及并行度等。

1. Topology运行模式
2. Topology运行流程
3. Topology的方法调用流程
4. Topology并行度
5. 终止Topology
6. Topology跟踪