1. Storm并行相关的概念

Storm集群有很多节点，按照类型分为nimbus（主节点）、supervisor（从节点）。 在conf/storm.yaml中配置了一个supervisor有多个槽（supervisor.slots.ports），每个槽就是一个JVM，就是一个worker，在每个worker里面可以运行多个线程叫做executor，在executor里运行一个topology的一个component（spout、bolt）叫做task。

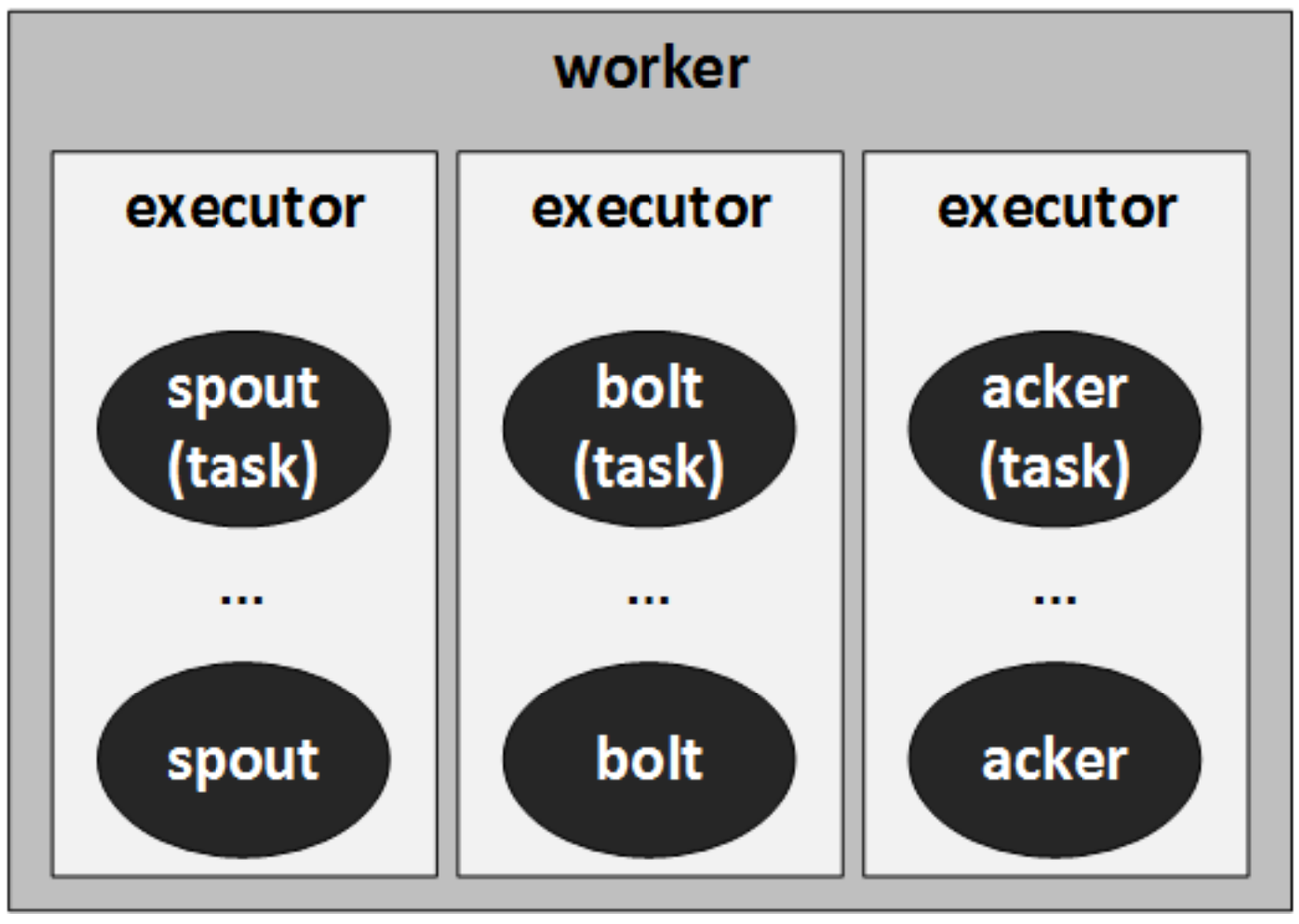
1). 一个Storm集群有一个nimbus进程
2). 每个Storm集群机器有一个supervisor进程
3). 每个supervisor进程配置有多个JVM槽，每个JVM槽称为worker
4). 一个Storm拓扑在一个worker（JVM）运行
5). 每个worker可以运行多个线程，称为executor
6). 每个executor执行拓扑的一个component（spout或bolt），称为task

2. Storm并行配置

1). supervisor是storm集群配置的，执行storm supervisor时，产生一个supervisor节点。
2). worker进程是在storm/conf/storm.yaml文件中选项supervisor.slots.ports配置的。
worker进程数量也可以通过config.setNumWorkers(workers)设置。
3). executor是通过builder.setSpout(id, spout, parallelism_hint)和builder.setBolt(id, bolt, parallelism_hint)设置的。
4). task是通过boltDeclarer.setNumTasks(num)设置的。
默认情况下，每个supervisor启动4个worker，每个worker启动1个executor，每个executor中会有1个task。

例如：
topologyBuilder.setBolt("green-bolt", new GreenBolt(), 2) .setNumTasks(4) .shuffleGrouping("blue-spout);
上面指定使用2个executor、4个task运行green-bolt，Storm会使用2个executor，每个executor运行2个task来运行green-bolt

3. 图文详解



1). worker即进程，一个worker就是一个进程，进程里面包含一个或多个线程
一个worker处理topology的一个子集，同一个子集可被多个worker同时处理，一个worker有且仅为一个topology服务，不会存在一个worker即处理topology1的几个节点，又处理topology2的几个节点
2). 一个线程就是一个executor，一个线程会处理一个或多个任务
一个executor处理一个节点，但这个节点可能会有多个实例对象，所以可通过配置并发度和setNumTask来配置一个executor同时处理多少个task。默认情况下task的数目等于executor线程数目，即1个executor线程只运行1个task。
3). 一个任务就是一个task，一个task就是一个节点类的实例对象
默认情况下一个executor就处理一个task。如果处理多个task，executor会循环遍历执行task。

一个topology的提交过程：

1. 非本地模式下，客户端通过thrift调用nimbus接口，来上传代码到nimbus并触发提交操作.
2. nimbus进行任务分配，并将信息同步到zookeeper.
3. supervisor定期获取任务分配信息，如果topology代码缺失，会从nimbus下载代码，并根据任务分配信息，同步worker.
4. worker根据分配的tasks信息，启动多个executor线程，同时实例化spout、bolt、acker等组件，此时，等待所有connections（worker和其它机器通讯的网络连接）启动完毕，此storm-cluster即进入工作状态。
5. 除非显示调用kill topology，否则spout、bolt等组件会一直运行。