Storm 实习报告

111172 徐鸿飞

1. 摘要

Apache Storm 是一个免费的开源分布式实时计算系统。Apache Storm 使得可靠处理无限数据流变得容易，就像实时处理 Hadoop 进行批处理一样。Apache Storm 很简单，可以与任何编程语言一起使用，并且使用起来很有趣！

本次作业是 Storm 编程实践，主要涉及的是 Storm 的安装与配置，简单的本地 storm 的调试运行和将 Word CountTopology 提交到 storm 中运行。

2. 目的

了解 storm 的运行机制，学习编写 storm 的 Topology 并打包提交到 storm 中运行。

3. 软硬件环境

平台	VMware 15 WorkStation
硬件	2h4g centos 7
软件	JAVA 8
	Hadoop 2.10.0
	HBase 1.6.0
	Zookeeper 3.6.1
	Storm 2.1.0
IDE	IDEA

4. 数据量

一些简单的测试数据。

5. 方法/算法

WordCount -Storm 编程实践：

Storm 主要术语包括 Streams、Spouts、Bolts、Topology 和 Stream Groupings；

主要用到的有：

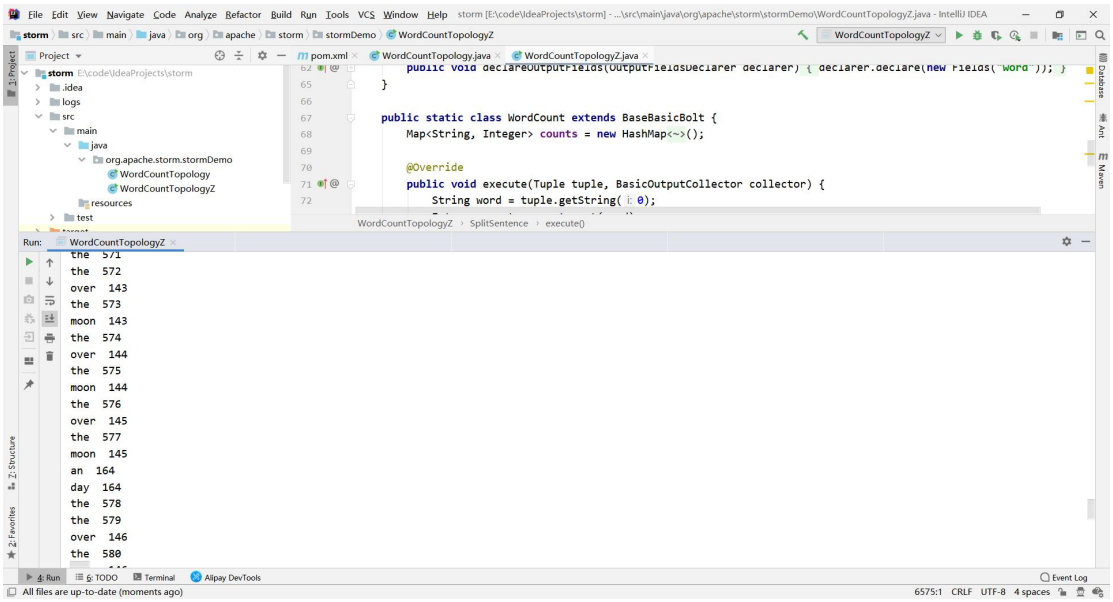
Spout 流的源头，比如从 Kafka 读取。

Bolt 处理输出流，同时产生新的流。

Topology 由 Spout，Bolt 构成的网络。

本示例中，RandomSentenceSpout 发射出一个个的句子，SplitSentence 进行分词，然后发射出一个个的单词，WordCount 进行单次计数。由此组成了一个链。而在实际的分布式环境中，会由 Stream Grouping 决定一个 Tuple 路由到哪个 Task 去执行。

本地调试运行：



部署到 storm 上：

先要安装 storm， Storm 使用 Zookeeper 来作为分布式协调组件，负责 Nimbus 和多个 Supervisor 之间的所有协调工作。借助于 Zookeeper，若 Nimbus 进程或 Supervisor 进程意外终止，重启时也能读取、恢复之前的状态并继续工作，使得 Storm 极其稳定。所以在这之前需要先安装 zookeeper；下载 zookeeper，要注意的是最新版本的 zookeeper 应该下载 apache-zookeeper-3.6.1-bin.tar.gz，而不是原先的 apache-zookeeper-3.6.1.tar.gz

开启 nimbus：

```
[root@ali apache-storm-2.1.0]# bin/storm nimbus
Running: /usr/local/java/jdk1.8.0_251/bin/java -server -Ddaemon.name=nimbus -Dstorm.options= -Dstorm.home=/usr/bigdata/apache-storm-2.1.0 -Dstorm.log.dir=/usr/bigdata/apache-storm-2.1.0/logs -Djava.library.path=/usr/local/lib:/opt/local/lib:/usr/lib:/usr/lib64 -Dstorm.conf.file= -cp /usr/bigdata/apache-storm-2.1.0/*:/usr/bigdata/apache-storm-2.1.0/lib/*:/usr/bigdata/apache-storm-2.1.0/extlib/*:/usr/bigdata/apache-storm-2.1.0/extlib-daemon/*:/usr/bigdata/apache-storm-2.1.0/conf -Xmx1024m -Djava.deserialization.disabled=true -Dlogfile.name=nimbus.log -Dlog4j.configurationFile=/usr/bigdata/apache-storm-2.1.0/log4j2/cluster.xml org.apache.storm.daemon.nimbus.Nimbus
```

开启 supervisor：

```
[root@ali ~]# cd /usr/bigdata/apache-storm-2.1.0/
[root@ali apache-storm-2.1.0]# bin/storm supervisor&
[1] 20094
[root@ali apache-storm-2.1.0]# Running: /usr/local/java/jdk1.8.0_251/bin/java -server -Ddaemon.name=supervisor -Dstorm.options= -Dstorm.home=/usr/bigdata/apache-storm-2.1.0 -Dstorm.log.dir=/usr/bigdata/apache-storm-2.1.0/logs -Djava.library.path=/usr/local/lib:/opt/local/lib:/usr/lib:/usr/lib64 -Dstorm.conf.file= -cp /usr/bigdata/apache-storm-2.1.0/*:/usr/bigdata/apache-storm-2.1.0/lib/*:/usr/bigdata/apache-storm-2.1.0/extlib/*:/usr/bigdata/apache-storm-2.1.0/extlib-daemon/*:/usr/bigdata/apache-storm-2.1.0/conf -Xmx256m -Djava.deserialization.disabled=true -Dlogfile.name=supervisor.log -Dlog4j.configurationFile=/usr/bigdata/apache-storm-2.1.0/log4j2/cluster.xml org.apache.storm.daemon.supervisor.Supervisor
```

Jps：

```
[root@ali ~]# jps
31399 jar
14457 QuorumPeerMain
5721 Nimbus
5371 Jps
20094 Supervisor
```

每次开启服务都要创建一个新的终端；

Worker 进程:每个 worker 进程都属于一个特定的 Topology，每个 Supervisor 节点的 worker 可以有多个，每个 worker 对 Topology 中的每个组件（Spout 或 Bolt）运行一个或者多个 executor 线程来提供 task 的运行服务；所有 Topology 任务的提交必须在 Storm 客户端节点上进行，提交后，由 Nimbus 节点分配给其他 Supervisor 节点进行处理；

Nimbus 节点首先将提交的 Topology 进行分片，分成一个个 Task，分配给相应的 Supervisor，并将 Task 和 Supervisor 相关的信息提交到 Zookeeper 集群上。

6. 结果和分析

在 storm 中提交一个 Topology 很方便，在客户端执行 storm jar xxx.jar com.xx.yy.TheTopology topologyName 即可，这个命令由 python 实现，在这儿会调用 exec_storm_class 并组装出 java 命令，并调用。jarfile 即是 jar 在客户端本地文件。

这里相当于执行了 jar 里面 Topology 的 Main 方法，而 Main 方法里面一般会调用 storm 的 api，所以，客户端只是提供了一个提交途径而已，最终会回到 api 进行处理。

7. 结论

Storm 可用于许多领域中，如实时分析、在线机器学习、持续计算、远程 RPC、数据提取加载转换等，可以大大增加实时数据的价值，为业务分析带来质的提升;Storm 流处理框架具有可扩展性、高容错性、能可靠地处理消息的特点，使用简单，学习和开发成本较低。Storm 框架对设计概念进行了抽象化，其主要术语包括 Streams、Spouts、Bolts、Topology 和 Stream Groupings，在 Topology 中定义整体任务的处理逻辑，再通过 Bolt 具体执行，Stream Groupings 则定义了 Tuple 如何在不同组件间进行传输。