

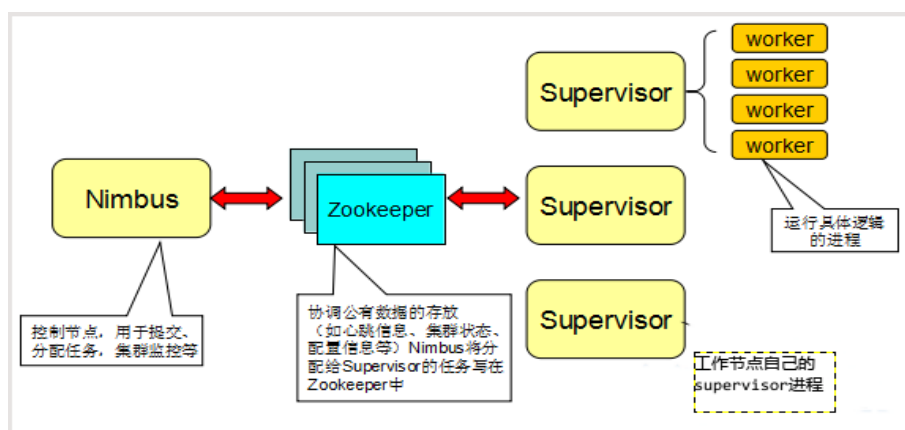
Storm 集群安装部署步骤

1. Storm 集群组件

Storm 集群中包含两类节点：主控节点（Master Node）和工作节点（Work Node）。其分别对应的角色如下：

主控节点（Master Node）上运行一个被称为 Nimbus 的后台程序，它负责在 Storm 集群内分发代码，分配任务给工作机器，并且负责监控集群运行状态。Nimbus 的作用类似于 Hadoop 中 JobTracker 的角色。

每个工作节点（Work Node）上运行一个被称为 Supervisor 的后台程序。Supervisor 负责监听从 Nimbus 分配给它执行的任务，据此启动或停止执行任务的工作进程。每一个工作进程执行一个 Topology 的子集；一个运行中的 Topology 由分布在不同工作节点上的多个工作进程组成。



Storm 集群组件

Nimbus 和 Supervisor 节点之间所有的协调工作是通过 Zookeeper 集群来实现的。此外，Nimbus 和 Supervisor 进程都是快速失败（fail-fast）和无状态（stateless）的；Storm 集群所有的状态要么在 Zookeeper 集群中，要么存储在本地磁盘上。这意味着你可以用 kill -9 来杀死 Nimbus 和 Supervisor 进程，它们在重启后可以继续工作。这个设计使得 Storm 集群拥有不可思议的稳定性。

2. 安装 Storm 集群

现在详细描述如何搭建一个 Storm 集群。下面是接下来需要依次完成的安装步骤：

- 搭建 Zookeeper 集群；
- 安装 Storm 依赖库；
- 下载并解压 Storm 发布版本；
- 修改 storm.yaml 配置文件；
- 启动 Storm 各个后台进程。

2.1 搭建 Zookeeper 集群

Storm 使用 Zookeeper 协调集群，由于 Zookeeper 并不用于消息传递，所以 Storm 给 Zookeeper 带来的压力相当低。大多数情况下，单个节点的 Zookeeper 集群足够胜任，不过为了确保故障恢复或者部署大规模 Storm 集群，可能需要更大规模节点的 Zookeeper 集群（对于 Zookeeper 集群的话，官方推荐的最小节点数为 3 个）。在 Zookeeper 集群的每台机

器上完成以下安装部署步骤：以下所有操作均需在每个节点上完成或者将一个配置好的节点复制到另一个节点上。

1) 下载安装 Java JDK，官方下载链接为

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>，JDK 版本为 JDK 6 或以上，本次是 java1.7，若机器已安装，跳过此步。

2) 下载后解压安装 Zookeeper 包，下载最近的一个稳定版，官方下载链接为：www-us.apache.org/dist/zookeeper/zookeeper-3.4.8/zookeeper-3.4.8.tar.gz

3) 解压 zookeeper。

```
[root@master ~] tar -zxvf zookeeper-3.4.8.tar.gz
[root@master ~] mv zookeeper-3.4.8 zookeeper
```

3) 根据 Zookeeper 集群节点情况，创建如下格式的 Zookeeper 配置文件

```
[root@master ~] vim zoo.cfg:
tickTime=2000
dataDir=/root/zookeeper/
clientPort=2181
initLimit=5
syncLimit=2
server.1=master:2888:3888
server.2=slave1:2888:3888
server.3=slave2:2888:3888
```

其中，dataDir 指定 Zookeeper 的数据文件目录；其中 server.id=host:port:port，id 是为每个 Zookeeper 节点的编号，保存在 dataDir 目录下的 myid 中，master, slave1, slave2, 表示各个 Zookeeper 节点的 hostname，第一个 port 是用于连接 leader 的端口，第二个 port 是用于 leader 选举的端口。

4) 在 dataDir 目录下创建 myid 文件，文件中只包含一行，且内容为该节点对应的 server.id 中的 id 编号。

5) 将 zookeeper 加入系统环境变量，：

```
[root@master ~] vim /etc/profile
export ZOOKEEPER_HOME=/root/zookeeper
export PATH=$PATH:$ZOOKEEPER/bin
```

也可以通过 bin/zkServer.sh 脚本启动 Zookeeper 服务。

6) 启动 Zookeeper 服务，在各个节点上启动 zookeeper 服务。

```
[root@master ~] zkServer.sh start
JMX enabled by default
Using config: /root/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@slave1 ~] zkServer.sh start
[root@slave2 ~] zkServer.sh start

[root@master ~]# jps
2458 Jps
2412 QuorumPeerMain
```

QuorumPeerMain 为 zookeeper 的 java 守护进程，可以看到已经启动了，在 slave1 和 slave2 上和 master 上一样，通过 Zookeeper 客户端测试服务是否可用：

```
[root@slave1] zkCli.sh -server master:2181
```

客户端下，进入 zookeeper/src/c 目录下，编译单线程或多线程客户端：

```
[root@slave1] ./
```

```
make cli_st
```

```
make cli_mt
```

运行进入 C 客户端：

```
cli_st 127.0.0.1:2181
```

```
cli_mt 127.0.0.1:2181
```

至此，完成了 Zookeeper 集群的部署与启动。

注意事项：

1. 由于 Zookeeper 是快速失败 (fail-fast) 的，且遇到任何错误情况，进程均会退出，因此，最好能通过监控程序将 Zookeeper 管理起来，保证 Zookeeper 退出后能被自动重启。这个地方可以自己写个脚本。。
2. Zookeeper 运行过程中会在 dataDir 目录下生成很多日志和快照文件，而 Zookeeper 运行进程并不负责定期清理合并这些文件，导致占用大量磁盘空间，因此，需要通过 cron 等方式定期清除没用的日志和快照文件。具体命令格式如下：java -cp zookeeper.jar:log4j.jar:conf org.apache.zookeeper.server.PurgeTxnLog <dataDir> <snapDir> -n <count>

2.2 安装 Storm 依赖库

接下来，需要在 Nimbus 和 Supervisor 机器上安装 Storm 的依赖库，具体如下：

1. ZeroMQ 4.0.7 - 请勿使用 2.1.10 版本，因为该版本的一些严重 bug 会导致 Storm 集群运行时出现奇怪的问题。少数用户在 2.1.7 版本会遇到 "IllegalArgumentException" 的异常，此时降为 2.1.4 版本可修复这一问题。
2. JZMQ
3. Python 2.6.6
4. unzip
5. tar

以上依赖库的版本是经过 Storm 测试的，Storm 并不能保证在其他版本的 Java 或 Python 库下可运行。

2.2.1 安装 ZMQ 2.1.7

下载后编译安装 ZMQ：

```
[root@master ~]# wget http://download.zeromq.org/zeromq-4.0.7.tar.gz
```

```
[root@master ~]# tar -xvzf zeromq-4.0.7.tar.gz
```

```
[root@master ~]# cd zeromq-4.0.7
```

```
[root@master ~]# ./configure
```

```
[root@master ~]# make
```

```
[root@master ~]# sudo make install
```

注意事项：

1. 如果安装过程报错 uuid 找不到，则通过如下的包安装 uuid 库：

```
[root@master ~]# yum install e2fsprogs1 -b current
```

```
[root@master ~]# yum install e2fsprogs-devel -b current
```

2.2.2 安装 JZMQ

下载后编译安装 JZMQ：

```
[root@master ~]# git clone https://github.com/zeromq/jzmq.git
```

```
[root@master ~]# cd jzmq
```

```
[root@master ~]# ./autogen.sh
```

```
[root@master ~]# ./configure
```

```
[root@master ~]# make
```

```
[root@master ~]# sudo make install
```

为了保证 JZMQ 正常工作，可能需要完成以下配置：

1. 正确设置 JAVA_HOME 环境变量
2. 安装 Java 开发包
3. 升级 autoconf

2.2.3 安装 Python2.6 以上

1. 下载 Python2.7.6:

```
[root@master ~] wget https://www.python.org/ftp/python/2.7.6/Python-2.7.6.tgz
```

2. 编译安装 Python2.7.6:

```
[root@master ~]# tar -jxvf Python-2.7.6.tgz
[root@master ~]# cd Python-2.6.6
[root@master ~]# ./configure
[root@master ~]# make
[root@master ~]# make install
```

3. 测试 Python2.7.6:

```
[root@master ~]# python -v
[root@master ~]# python 2.7.6
```

2.3 下载并解压 Storm 发布版本

1. 下载 Storm 发行版本，推荐使用 Storm-0.10.0:

下载地址：<http://storm.apache.org/downloads.html>

2. 解压到安装目录下:

```
[root@master ~]# tar -cxvf apache-storm-0.10.0.tar.gz
[root@master ~]# mv apache-storm-0.10.0 storm
```

2.4 将 Storm 的安装路径加入系统环境变量

```
[root@master ~]# vim /etc/profile
export STORM_HOME=/root/storm
export PATH=$PATH:$STORM_HOME/bin
```

2.5 修改 storm.yaml 配置文件

Storm 发行版本解压目录下有一个 conf/storm.yaml 文件，用于配置 Storm。conf/storm.yaml 中的配置选项将覆盖 defaults.yaml 中的默认配置。以下配置选项是必须在 conf/storm.yaml 中进行配置的：

```
[root@master ~]# vim storm/conf/storm.yaml
storm.zookeeper.servers:
  - "master"
  - "slaver1"
  - "slaver2"
storm.zookeeper.port: 2181
nimbus.host: "master"

storm.local.dir: /root/storm/data
```

说明：

- 1) **storm.zookeeper.servers**: Storm 集群使用的 Zookeeper 集群地址，其格式如下：
- 2) **storm.local.dir**: Nimbus 和 Supervisor 进程用于存储少量状态，如 jars、confs 等的本地磁盘目录，需要提前创建该目录并给以足够的访问权限。然后在 storm.yaml 中配置该目录

3) **nimbus.host**: Storm 集群 Nimbus 机器地址, 各个 Supervisor 工作节点需要知道哪个机器是 Nimbus, 以便下载 Topologies 的 jars、confs 等文件。

4) **supervisor.slots.ports**: 对于每个 Supervisor 工作节点, 需要在 storm.yaml 中配置该工作节点可以运行的 worker 数量。每个 worker 占用一个单独的端口用于接收消息, 该配置选项即用于定义哪些端口是可被 worker 使用的。默认情况下, 每个节点上可运行 4 个 workers, 分别在 6700、6701、6702 和 6703 端口, 如:

```
supervisor.slots.ports:
```

- 6700
- 6701
- 6702
- 6703

2.6 启动 Storm 各个后台进程

最后一步, 启动 Storm 的所有后台进程。和 Zookeeper 一样, Storm 也是快速失败 (fail-fast) 的系统, 这样 Storm 才能在任意时刻被停止, 并且当进程重启后被正确地恢复执行。这也是为什么 Storm 不在进程内保存状态的原因, 即使 Nimbus 或 Supervisors 被重启, 运行中的 Topologies 不会受到影响。

以下是启动 Storm 各个后台进程的方式: (注意, 启动 storm 之前必须先启动 zookeeper 服务进程。

1. **Nimbus**: 在 Storm 主控节点上运行 "bin/storm nimbus >/dev/null 2>&1 &" 启动 Nimbus 后台程序, 并放到后台执行;

```
[root@master ~]# storm nimbus >/dev/null 2>&1 &
```

2. **Supervisor**: 在 Storm 各个工作节点上运行 "bin/storm supervisor >/dev/null 2>&1 &" 启动 Supervisor 后台程序, 并放到后台执行;

```
[root@slave1 ~]# storm supervisor >/dev/null 2>&1 &
```

```
[root@slave2 ~]# storm supervisor >/dev/null 2>&1 &
```

3. **UI**: 在 Storm 主控节点上运行 "bin/storm ui >/dev/null 2>&1 &" 启动 UI 后台程序, 并放到后台执行, 启动后可以通过 `http://{nimbus host}:8080` 观察集群的 worker 资源使用情况、Topologies 的运行状态等信息。

```
[root@master ~]# storm ui >/dev/null 2>&1 &
```

2.7 查看集群守护进程状态。

以下是正常的守护进程状态, 其中在 master 上有 3 个守护进程, 在 slaves 上有 2 个守护进程, zookeeper 对应的进程是 QuorumPeerMain, 主控进程是 nimbus, 工作节点维护进程是 supervisor, core 进程是系统监控与管理界面服务进程。

```
[root@master ~]# jps
2916 nimbus
2890 QuorumPeerMain
3088 Jps
2999 core
[root@slave1 ~]# jps
1820 QuorumPeerMain
1848 supervisor
1960 Jps
```

注意事项:

1. Storm 后台进程被启动后, 将在 Storm 安装部署目录下的 logs/子目录下生成各个进程的日志文件。
2. 经测试, Storm UI 必须和 Storm Nimbus 部署在同一台机器上, 否则 UI 无法正常工作。

作，因为 UI 进程会检查本机是否存在 Nimbus 链接。

3. 为了方便使用，可以将 bin/storm 加入到系统环境变量中。

至此，Storm 集群已经部署、配置完毕，可以向集群提交拓扑任务运行了。

3. 向集群提交任务

1) 启动 Storm Topology:

```
[root@master ~]# storm jar mycode.jar MyTopology arg1 arg2 arg3
```

其中，mycode.jar 是包含 Topology 实现代码的 jar 包，MyTopology 的 main 方法是 Topology 的入口，arg1、arg2 和 arg3 为 MyTopology 执行时需要传入的参数。

2) 停止 Storm Topology:

```
[root@master ~]# storm kill MyTopology
```

其中，MyTopology 为 Topology 提交到 Storm 集群时指定的 Topology 任务名称。