Storm实战常见问题及解决方案

问题锦集1 关于Storm集群1.1 关于storm集群的环境变量配置问题

安装好JDK后，需要配置环境变量，通常情况下出于经验，我们往往会修改/etc/profile的值进行环境变量配置，但这在安装JDK以及后面安装的storm集群、zookeeper集群以及metaq集群时会出问题，这时候我们需要在/etc/.bashrc文件中加入环境变量，不然安装的java和ZK集群等就无法使用，尤其这个问题在我用shell写调度脚本去启动storm集群的时候就遇到过，如果没有将java的环境变量配置在/etc/.bashrc文件中，就会报一个错，这个问题在后面我会提到。

1.2 关于zookeeper集群安装问题

记得刚刚接触storm，在安装zookeeper集群的时候有这样的考虑：为什么不可以把zookeeper只安装在nimbus上，然后让其他的supervisor来它这里读取任务？如果在每台机器上都安装zookeeper，那nimbus分配任务的时候，是每台机器上的zookeeper都收到同一份的任务，还是只是将分配给每个supervisor节点的那部分写到同一节点上的zookeeper中？

有朋友解答说：ZK也是以集群的方式工作的，ZK集群内部有他自己的一套相互通信机制，而storm正是要借助其通讯机制，例如任务下发等，往往在执行一个任务的时候，storm会把任务及相关执行的代码经过序列化之后发送到各个ZK节点供supervisor去下载，然后才会各自执行自己部分的代码或者任务。说的直接一点就是每个ZK节点收到的任务是一样的，而supervisor只需要下载属于自己的任务即可。

1.3 关于Storm中tuple 的可靠处理问题

Storm 为了保证tuple 的可靠处理，需要保存tuple 信息，这样会不会导致内存泄漏？

关于这个问题，其实网上是有资料进行了详细的解释的。这里只是大概将一下，如果还不明白，可以上网搜搜“storm可靠处理”。Storm 为了保证tuple 的可靠处理，acker 会保存该节点创建的tuple id的xor （异或）值，这个值称为ack value，那么每ack 一次，就将tuple id 和ack value做异或(xor)。当所有产生的tuple 都被ack 的时候，ack value 必定为0。这是个很简单的策略，对于每一个tuple 也只要占用约20 个字节的内存。对于100万tuple，也才20M 左右，所以一般情况下是不用考虑内存泄漏问题的。

1.4 关于storm计算结果的存放问题

很多人在刚刚学习Storm 的时候都会有这个问题：storm处理后的结果保存在哪里？ 内存中？还是其他地方？

官方解释说： Storm 是不负责保存计算结果的，这是应用程序里需要负责的事情，如果数据不大，你可以简单地保存在内存里，也可以每次都更新数据库，也可以采用NoSQL存储。storm 并没有像s4 那样提供一个Persist API，根据时间或者容量来做存储输出。这部分事情完全交给用户。数据存储之后的展现，也是你需要自己处理的，storm UI 只提供对topology 的监控和统计。

1.5 关于Storm如何处理重复的tuple问题

有人问到Storm 是怎么处理重复的tuple？

因为Storm 要保证tuple 的可靠处理，当tuple 处理失败或者超时的时候，spout 会fail 并重新发送该tuple，那么就会有tuple 重复计算的问题。这个问题是很难解决的，storm 也没有提供机制帮助你解决。不过也有一些可行的策略：

（1）不处理，这也算是种策略。因为实时计算通常并不要求很高的精确度，后

续的批处理计算会更正实时计算的误差。

（2）使用第三方集中存储来过滤，比如利用MySQL、MemCached 或者Redis 根据逻辑主键来去重。

（3）使用bloom filter 做过滤，简单高效。

1.6 关于task与executor的关系问题

在storm的学习过程中，有许多人问到task与executor的关系问题。

在我们安装配置storm的时候，不知大家是否主要到了一个问题，就是我们在配置的时候会加几个worker的端口( supervisor.slots.ports:)，比如众多文档中提到的6700/6701等等类似的东西。没错，这就是我们定义了该supervisor最多的worker数，worker中执行一个bolt或者spout线程，我们就称之为task，而executor是物理上的线程概念，我们可以将其称为执行线程；而task更多是逻辑概念上的，有时候bolt与spout的task会共用一个executor，特别是在系统负荷比较高的时候。

1.7 关于Storm UI显示内容的问题

Storm UI 里spout 统计的complete latency 的具体含义是什么？为什么emit 的数目会是acked 的两倍？

简单地说，complete latency 表示了tuple 从emit 到被acked 经过的时间，可以认为是tuple 以及该tuple 的后续子孙（形成一棵树）整个处理时间。其次spout 的emit 和transfered 还统计了spout 和acker 之间内部的通信信息，比如对于可靠处理的spout 来说，会在emit 的时候同时发送一个\_ack\_init 给acker，记录tuple id 到task id 的映射，以便ack 的时候能找到正确的acker task。

1.8 关于Storm的ack和fail问题

在学习storm的过程中，有不少人对storm的Spout组件中的ack及fail相关的问题存在困惑，这里做一个简要的概述。

Storm保证每一个数据都得到有效处理，这是如何保证的呢？正是ack及fail机制确保数据都得到处理的保证，但是storm只是提供给我们一个接口，而具体的方法得由我们自己来实现。例如在spout下一个拓扑节点的bolt上，我们定义某种情况下为数据处理失败，则调用fail，则我们可以在fail方法中进行数据重发，这样就保证了数据都得到了处理。其实，通过读storm的源码，里面有讲到，有些类（BaseBasicBolt？）是会自动调用ack和fail的，不需要我们程序员去ack和fail，但是其他Bolt就没有这种功能了。

1.9 关于IRichBolt与IBasicBolt接口的区别

首先从类的组成上进行分析可以看到，IBasicBolt接口只有execute方法和declareOutputFields方法，而IRichBolt接口上除了以上几个方法还有prepare方法和cleanup及map方法。而且其中execute方法是有些不一样的，其参数列表不同。

总体来说Rich方法比较完善，我们可以使用prepare方法进行该Bolt类的初始化工作，例如我们链接数据库时，需要进行一次数据库连接操作，我们就可以把该操作放入prepare中，只需要执行一次就可以了。而cleanup方法能在该类调用结束时进行收尾工作，往往在处理数据的时候用到，例如在写hdfs（hadoop的文件系统）数据的时候，在结束时需要进行数据clear，则需要进行数据收尾。当然，根据官网及实际的测验，该方法往往是执行失败的。

2 关于Topology发布2.1 发布topologies 到远程集群时，出现Nimbus host is not set 异常

原因是Nimbus 没有被正确启动起来，可能是storm.yaml 文件没有配置，或者配置有问题。

解决方法：打开storm.yaml 文件正确配置：nimbus.host: "xxx.xxx.xxx.xxx"，重启nimbus 后台程序即可。

2.2 发布topology到远程集群时，出现AlreadyAliveException(msg: xxx is already active)异常

原因是提供的topology 与已经在运行的topology 重名。

解决方法：发布时换一个拓扑名称即可。

2.3 启动Supervisor 时，出现java.lang.UnsatisfiedLinkError

具体信息：启动Supervisor 时，出现java.lang.UnsatisfiedLinkError:

/usr/local/lib/libjzmq.so.0.0.0: libzmq.so.1: cannot open shared object

file: No such file or directory 异常。

原因是未找到zmq 动态链接库。

解决方法1：配置环境变量 export LD\_LIBRARY\_PATH=/usr/local/lib

解决方法2：编辑/etc/ld.so.conf 文件，增加一行：/usr/local/lib。再执行

sudo ldconfig 命令，重启Supervisor。

2.4 发布topologies 时，出现不能序列化log4j.Logger 的异常

原因是日志系统无法正确支付序列化。

解决方法：使用slf4j 代替log4j。

2.5 bolt 在处理消息时，worker 的日志中出现Failing message

原因：可能是因为Topology 的消息处理超时所致。

解决方法：提交Topology 时设置适当的消息超时时间，比默认消息超时时间（30

秒）更长。比如：

conf.setMessageTimeoutSecs(60);

2.6 在打包toplogy工程的时候, 如果采用assembly方式, 对于相关的依赖的配置一般要这样:

Xml代码

1. <dependencySets>

2. <dependencySet>

3. <outputDirectory>/</outputDirectory>

4. <unpack>true</unpack>

5. <excludes>

6. <exclude>storm:storm</exclude>

7. </excludes>

8. </dependencySet>

9. </dependencySets>

wiki上说可以用<scope>compile</scope>。然后将storm依赖设置为runtime, 貌似不行。 另外就是所有的依赖包将全部解压, 然后将所有依赖的配置和class文件生成一个文件。这个是通过<unpack>true</unpack>参数来控制的。

2.7 在提交topology的时候有时可能出现如下异常:

Exception in thread "main" java.lang.IllegalArgumentException: Nimbus host is not set  
at backtype.storm.utils.NimbusClient.<init>(NimbusClient.java:30)  
at backtype.storm.utils.NimbusClient.getConfiguredClient(NimbusClient.java:17)  
at backtype.storm.StormSubmitter.submitJar(StormSubmitter.java:78)  
at backtype.storm.StormSubmitter.submitJar(StormSubmitter.java:71)  
at backtype.storm.StormSubmitter.submitTopology(StormSubmitter.java:50)  
at com.taobao.kaleidoscope.storm.IcdbTopology.main(IcdbTopology.java:59)

但是启动nimbus是没有问题的, 这个主要因为conf\_dir路径设置不正确, 在bin/storm脚本中需要加上这样一句:

Python代码

1. CONF\_DIR = STORM\_DIR + "/conf"

3 关于DRPC3.1 发布drpc 类型的topologies 到远程集群时，出现空指针异常，连接drpc服务器失败

原因是未正确配置drpc 服务器地址。

解决方法：在conf/storm.yaml 文件中增加drpc 服务器配置，启动配置文件中

指定的所有drpc 服务。内容如下：

drpc.servers:

- "drpc 服务器ip"

3.2 客户端调用drpc 服务时，worker 的日志中出现Failing message，而bolt都未收到数据

错误日志如下所示：

2011-12-02 09:59:16 task [INFO] Failing message

backtype.storm.drpc.DRPCSpout$DRPCMessageId@3770bdf7: source: 1:27,

stream: 1, id: {-5919451531315711689=-5919451531315711689},

[foo.com/blog/1, {"port":3772,"id":"5","host":"10.0.0.24"}]

原因是主机名，域名，hosts 文件配置不正确会引起这类错误。

解决方法：检查并修改storm 相关机器的主机名，域名，hosts 文件。重启网络服务：service network restart。重启storm，再次调用drpc 服务，成功。Hosts 文件中必须包含如下

内容：

[nimbus 主机ip] [nimbus 主机名] [nimbus 主机别名]

[supervisor 主机ip] [supervisor 主机名] [supervisor 主机别名]

[zookeeper 主机ip] [zookeeper 主机名] [zookeeper 主机别名]

4 关于jzmq安装4.1 storm 启动时报no jzmq in java.library.path 错误

原因是找不到jzmq，默认情况下在执行install\_zmq.sh 时，那些.so 文件

安装路径在/usr/local/lib，但是实际安装时可能装在其他的路径下了。

解决方法：在storm.yaml 中添加：

java.library.path:

"/opt/storm/jzmq/lib:/opt/storm/zeromq/lib:/usr/local/lib:/opt/local/

lib:/usr/lib"

4.2 安装jzmq 时遇到No rule to make target ‘classdist\_noinst.stamp’的make 错误

具体的make 错误信息：

make[1]: \*\*\* No rule to make target `classdist\_noinst.stamp',needed by `org/zeromq/ZMQ.class'. Stop.

解决方法：手动创建classdist\_noinst.stamp 空文件。

touch src/classdist\_noinst.stamp

4.3 安装jzmq 时遇到cannot access org.zeromq.ZMQ 的make 错误

具体的make 错误信息：

error: cannot access org.zeromq.ZMQ class file for org.zeromq.ZMQ not found

javadoc: error - Class org.zeromq.ZMQ not found.

解决方法：手动编译，然后重新make 即可通过。

cd src

javac -d . org/zeromq/\*.java

cd ..

4.4 在部署storm节点的时候需要安装jzmq和zeromq, 在安装这两个依赖包之后, 需要执行sudo -u root ldconfig. 否则会出现异常:

2012-02-24 16:30:30 worker [ERROR] Error on initialization of server mk-worker  
java.lang.UnsatisfiedLinkError: /usr/local/lib/libjzmq.so.0.0.0: libzmq.so.1: cannot open shared object file: No such file or  
directory  
at java.lang.ClassLoader$NativeLibrary.load(Native Method)  
at java.lang.ClassLoader.loadLibrary0(ClassLoader.java:1803)  
at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1728)  
at java.lang.Runtime.loadLibrary0(Runtime.java:823)  
at java.lang.System.loadLibrary(System.java:1028)  
at org.zeromq.ZMQ.<clinit>(ZMQ.java:34)

5 关于Storm的配置问题

1. yaml跟我们一般用的属性配置文件有所不同, 它的要求更严格一些, 因此在往conf/storm.yaml中添加配置的时候必须注意，比如必须注意开始位置和冒号后面的空格, 否则配置不会生效。

2. 如何检查配置是否生效？

可以使用命令: storm localconfvalue 配置关键字

但是这个命令只能在nimbus上生效, 在supervisor看到的还是默认值. 不知道为什么 。

6 关闭storm相关进程6.1 关闭nimbus相关进程:

kill `ps aux | egrep '(daemon\.nimbus)|(storm\.ui\.core)' | fgrep -v egrep | awk '{print $2}'`

备注：这是在网上看到的，没有经过实际测试，有兴趣的朋友可以自己测试一下。

6.2 干掉supervisor上的所有storm进程:

kill `ps aux | fgrep storm | fgrep -v 'fgrep' | awk '{print $2}'`

备注：这是在网上看到的，没有经过实际测试，有兴趣的朋友可以自己测试一下。

7 关于Topology发布之后的log

1) 用storm jar ...将项目提交给storm集群后，想查看本项目的log信息，要到supervisor机器的：storm安装路径/logs/worker-number.log（其中的number视实际情况而定）中查看。

2) 如果是用daemontools启动的storm，daemontools监控的目录是/service/storm，那么到/service/storm/logs中查看worker-number.log日志。

3) 若要更改log的级别，是debug还是info等，在storm安装路径/log4j下有个配置文件，按需要修改即可。

4) Storm的debug模式下，它本身的log非常庞大，所以我觉得自己的代码中有些重要的信息，用info比较好，这样将storm的log级别调整为info比较方便查看。

8 关于maven打包问题8.1 首先maven的pom文件中的storm依赖，要么加exclude storm的相关语句（github有说明），要么加<scope>，如下：

<dependency>

<groupId>storm</groupId>

<artifactId>storm</artifactId>

<scope>test</scope>

</dependency>

加scope可以使打jar包时，不包含storm。如果包含了storm，那么提交到storm集群，会运行出错。官方要求打jar包时，要去除storm的依赖。

8.2 使用maven插件，在打jar包时，包含依赖。

在pom中加入：

<plugin>

<artifactId>maven-assembly-plugin</artifactId>

<configuration>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

<archive>

<manifest>

<mainClass>com.path.to.main.Class</mainClass>

</manifest>

</archive>

</configuration>

</plugin>

打jar包时使用命令：mvn assembly:assembly

8.3 依赖的jar冲突问题

如果本地依赖的jar与storm的lib下的jar有冲突，即都用了一个jar，但是版本不同，那么貌似目前只能改为跟storm保持统一。官方的讨论组是这样说的。

9 关于nimbus的启动问题9.1 Storm nimbus启动失败

在使用了storm一段时间后，需要重新部署storm的集群，主要是想将storm部署在其它机器上。做了以下错误操作：

1) 没有kill 正在运行的topology，kill nimbus和supervisor的storm进程

2) 删除了配置中"storm.local.dir"的文件夹内的内容

3) 启动storm nimbus

报错：  
backtype.storm.daemon.nimbus  
$fn\_\_2692$exec\_fn\_\_945\_\_auto\_\_\_\_2693$this\_\_2731@62135133  
java.io.FileNotFoundException: File '/opt/apps-install/storm/  
storm\_local/nimbus/stormdist/appFailed-6-1325065153/stormconf.ser'  
does not exist  
at  
org.apache.commons.io.FileUtils.openInputStream(FileUtils.java:137)  
at  
org.apache.commons.io.FileUtils.readFileToByteArray(FileUtils.java:  
1135)  
at backtype.storm.daemon.nimbus  
$read\_storm\_conf.invoke(nimbus.clj:128)  
at backtype.storm.daemon.nimbus  
$compute\_new\_task\_\_GT\_node\_PLUS\_port.invoke(nimbus.clj:244)  
at backtype.storm.daemon.nimbus  
$mk\_assignments.invoke(nimbus.clj:288)  
at backtype.storm.daemon.nimbus  
$fn\_\_2692$exec\_fn\_\_945\_\_auto\_\_\_\_2693$this\_\_2731.invoke(nimbus.clj:460)  
at backtype.storm.event$event\_manager  
$fn\_\_2068$fn\_\_2069.invoke(event.clj:25)  
at backtype.storm.event$event\_manager  
$fn\_\_2068.invoke(event.clj:22)  
at clojure.lang.AFn.run(AFn.java:24)  
at java.lang.Thread.run(Thread.java:662)  
2011-12-29 16:15:02 util [INFO] Halting process: ("Error when  
processing an event")  
报错原因：因为没有先kill topology，所以在启动nimbus时，zookeeper中依然保留了上次运行着的topology的信息。

解决办法：用zookeeper的zkCli.sh清理一下，我直接重装了zookeeper。但是据说在storm 0.6.1中解决了该bug。而我用的是storm 0.6.0。

10 Storm使用JVM参数

在配置文件storm.yaml中，有：

# to nimbus  
nimbus.childopts: "-Xmx1024m"

# to supervisor  
supervisor.childopts: "-Xmx1024m"

# to worker  
worker.childopts: "-Xmx768m"  
如果worker在运行时，需要用指定的JVM参数，那么可以像这样配置：  
worker.childopts: "-Dworker=worker -Xmx768m -Xdebug –Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt\_socket,address=8111,suspend=y,server=y "

11 关于spout/bolt的生命周期

一般来说spout/bolt的生命周期如下:

1 在提交了一个topology之后(在nimbus所在的机器), 创建spout/bolt实例(spout/bolt在storm中统称为component)并进行序列化；

2 将序列化的component发送给所有的任务所在的机器；

3 在每一个任务上反序列化component；

4 在开始执行任务之前, 先执行component的初始化方法(bolt是prepare, spout是open)；

因此component的初始化操作应该在prepare/open方法中进行, 而不是在实例化component的时候进行。

12关于storm与spring框架集成问题

首先声明一下，这个问题是当时有考虑到是否可以将storm与spring集成时，在网上看到的一点介绍，只是为了日后做参考。

在进行storm与spring集成时，本来想着一次就能成功，抱着很大的希望可是运行时竟然报了个java.io.NotSerializableException的异常。该异常要求被依赖注入的jar包实现序列化接口，但那些jar包都是别人开发的你不能一个一个都改掉源码才能用到项目里。

再网上找一下还真有人遇到类似的问题，具体原因是对storm的spout和bolt的生命周期理解的不够深刻。

一般来说spout/bolt的生命周期如下:

1.在提交了一个topology之后(在nimbus所在的机器), 创建spout/bolt实例(spout/bolt在storm中统称为component)并进行序列化.

2.将序列化的component发送给所有的任务所在的机器

3.在每一个任务上反序列化component.

4.在开始执行任务之前, 先执行component的初始化方法(bolt是prepare, spout是open).

因此component的初始化操作应该在prepare/open方法中进行, 而不是在实例化component的时候进行.

按照这种说法进行改造，结构该问题消失了。但接下来又有了新的问题：

Caused by: org.xml.sax.SAXParseException: Content is not allowed in prolog.

这个异常网上搜索之后发现原来是由于\*.xml文件编码的问题。原因是在从其他项目里或者编辑工具编辑时，在文件编码中加入了BOM头的原因，于是用notePad++打开xml文件选择去掉BOM头信息，重新进行保存即可。

13 关于java.lang.NoClassDefFoundError: clojure.core.protocols$

原因：JDK版本不匹配，安装虚拟机时系统自带一个jdk.1.5.0。

解决办法：检查jdk版本，卸载系统自带的JDK，使用自己安装的JDK版本。

＃　rpm –qa | grep java

# rpm –e –nodeps java-\*

配置环境变量，vi /etc/profile

重新执行一遍试试，貌似问题解决了。

14 关于storm连接Mysql

连接远程mysql是报如下错误：

message from server:"Host FILTER" is not allowed to connect to this MySQL server

解决方案：

很可能是你没有给其他IP访问你数据库的权限，你可以试试：

在MySql数据库的主机上，在mysql命令行中输入以下命令：

grant all on \*.\* to root@'%' identified by "111111" ;

这样，给任何IP都赋予了访问的权限，

任何IP都能以，用户名：root ，密码：111111

来进行局域网的访问！

(命令中\*.\*是通配任何IP，你也可以指定IP)

15 关于metaq启动的出现服务拒绝连接的问题

解决办法：在metaq安装目录下，删掉之前的日志文件，测试网络是否正常连接。将之前的服务的metaq进程kill掉，然后重启。

16 关于topology的spout与bolt

之前有问到，一个topology中可不可以有多个spout？这个问题貌似很幼稚啊，呵呵。关于这个问题，我是这样考虑的：实际应用中，如果我们每一条应用都创建一个topology的话，未免也太夸张了。如果是同一个应用，同一个数据来源，但是你想分几种方式对这个数据做处理的话，这时候就应该是建多个spout了，让这些spout并行去读数据，然后交给订阅这个spout的bolt去处理就行，没必要一种处理方式建一个topology。

17 关于shell脚本编码格式问题

这是我在写启动storm集群的shell脚本时遇到的一个实际问题。shell脚本运行时报错误：/bin/bash^M: bad interpreter

出现原因：windows上写的脚本，直接拷贝到linux系统上运行由于格式不兼容导致。

17.1 解决方案（一）：

1. 比如文件名为myshell.sh，vim myshell.sh

2. 执行vim中的命令 : set ff?查看文件格式，如果显示fileformat=dos，证明文件格式有问题。

3. 执行vim中的命令 :set fileformat=unix 将文件格式改过来就可以了，然后:wq保存退出就可以了。

17.2 解决方案（二）

或者使用最笨的方法：将windows下编辑好的脚本通过txt文本格式转换，然后在拷贝到linux下。

如果是使用Notepad编辑器进行编辑的话，可以在菜单栏上选择“编辑”—“档案格式转换”—“转换为 UNIX 格式”。

最后说明一下，这些问题只是storm应用过程中遇到的一小部分问题，其实还有很多问题是涉及到实际项目的考虑的，比如集群硬件要求，参数配置，日志处理等等，具体问题具体分析吧，也希望哪些在实际项目中用到storm的大神们，能多多和大家分享你们的实际经验，毕竟实践出真知，任何新技术，只有经过实际应用和实际检验，分享出来的东西才有说服力。