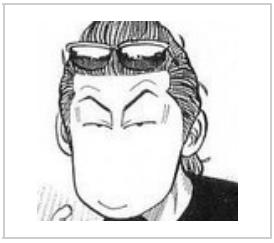

随枫霏

浏览: 16753 次

性别:

来自: 杭州

我现在离线

最近访客 更多访客>>


dylinshi126


zwj537


tanzhen-1988


sgq0085

文章分类

- 全部博客 (21)
- 工具 (4)
- memcached (1)
- CGLIB (1)
- 遇到的问题 (2)
- 设计模式 (3)
- logging (1)
- maven (2)
- 缓存 (1)
- ehcache (1)
- eclipse插件 (1)
- storm (2)

社区版块

- 我的资讯 (0)
- 我的论坛 (0)

Storm0.9.0集群搭建

博客分类: storm

storm

Storm集群可以搭建在AWS上,也可以直接手动部署在集群机器上。这里使用手动搭建的方式部署在一个机器上。

环境:ubuntu 13.10 64bit

1.Java安装

Java1.6的安装就不多的说了。

2.python安装

ubuntu自带了

3.zookeeper

单机模式部署方式

http://zookeeper.apache.org/doc/r3.3.3/zookeeperStarted.html#sc_InstallingSingleMode

Bash代码

```
1. wget http://apache.fayea.com/apache-mirror/zookeeper/zookeeper-3.4.5/zookeeper-3.4.5.tar.gz
2. chmod a+x zookeeper-3.4.5.tar.gz
3. tar zxvf zookeeper-3.4.5.tar.gz
```

进入conf目录下,新建zoo.cfg文件

Java代码

```
1. tickTime=2000
2. dataDir=/var/zookeeper
3. clientPort=2181
```

dataDir是zk用于存储文件的地址,确保运行用户有权限访问该目录

启动zk

Bash代码

```
1. bin/zkServer.sh start
```

- 我的问答 (1)

存档分类

- 2014-03 (2)
- 2014-01 (2)
- 2013-04 (1)
- 更多存档...

最新评论

1927105: th3639 写道第一步 下载 bug3271933.4.2h ...
运行TestNG出现CreateProcess error=206错误解决方法

wade6: panshunchang 写道文字写的不错,字体怎么这么小啊问 ...
学习分享——Memcached

panshunchang: 文字写的不错,字体怎么这么小啊 问一下,如果memcache ...
学习分享——Memcached

panshunchang: 文字写的不错,字体怎么这么小啊
学习分享——Memcached

wade6: 整理的不错,看着很舒服啊
学习分享——Memcached

验证一下

Java代码

```
1. echo ruok|nc localhost 2181
```

返回 imok说明zk起来了

4.ZeroMQ

http://download.zeromq.org/zeromq-3.2.2.tar.gz

Java代码

```
1. wget http://download.zeromq.org/zeromq-2.1.7.tar.gz
2. tar -xzf zeromq-2.1.7.tar.gz
3. cd zeromq-2.1.7
4. ./configure
5. make
6. sudo make install
```

期间可能遇到一些软件ubuntu没有安装 sudo apt-get install xxx 安装即可

5.JZMQ

Java代码

```
1. git clone https://github.com/nathanmarz/jzmq.git
2. cd jzmq
3. ./autogen.sh
4. ./configure
5. make
6. sudo make install
```

git的安装和配置

Java代码

```
1. sudo apt-get install git
2. git config --global user.name author #将用户名设为author
3. git config --global user.email author@corpmail.com #将用户邮箱设为author@corpmail.com
```

期间遇到的问题:

(1).make[1]: *** 没有规则可以创建“org/zeromq/ZMQ.class”需要的目标“classdist_noinst.stamp”。 停止

修正方法,创建classdist_noinst.stamp文件,

Java代码

```
1. touch src/classdist_noinst.stamp
```

(2).错误:无法访问 org.zeromq.ZMQ

修正方法,进入src目录,手动编译相关java代码

Java代码

```
1. javac -d . org/zeromq/*.java
```

6.Storm

下载最新release版本0.9.0.1

解压

Java代码

```
tar zxvf storm-0.9.0.1.tar.gz
```

修改Storm的配置文件 conf/storm.yaml

Java代码

```
storm.zookeeper.servers:
     - "localhost"
storm.local.dir: "/home/username/storm-0.9.0.1/workdir"
nimbus.host: "localhost"
```

说明:

storm.zookeeper.servers:这里使用的zk是本地的,所以用localhost

nimbus.host: 指明nimbus所在的机器

启动:

启动控制节点nimbus

Java代码

```
bin/storm nimbus >/dev/null 2>&1 &
```

启动任务节点supervisor

Java代码

```
bin/storm supervisor >/dev/null 2>&1 &
```

启动ui: 在nimbus节点上运行

Java代码

```
bin/storm ui >/dev/null 2>&1 &
```

UI启动后,可以通过 http://localhost:8080观察集群运行情况。

7.HelloWorld

这里使用«Getting started with Strom»书中的例子,这是一个word count 的例子

https://github.com/storm-book/examples-ch02-getting_started/zipball/master

需要修改一下pom文件

Xml代码

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
4.
5.          <modelVersion>4.0.0</modelVersion>
6.          <groupId>storm.book</groupId>
7.          <artifactId>Getting-Started</artifactId>
8.          <version>0.0.1-SNAPSHOT</version>
9.
10.         <build>
11.                 <plugins>
12.                         <plugin>
13.                                 <groupId>org.apache.maven.plugins</groupId>
14.                                 <artifactId>maven-compiler-plugin</artifactId>
15.                                 <version>2.3.2</version>
16.                                 <configuration>
17.                                         <source>1.6</source>
18.                                         <target>1.6</target>
19.                                         <compilerVersion>1.6</compilerVersion>
20.                                 </configuration>
21.                         </plugin>
22.                 </plugins>
23.         </build>
24.
25.         <repositories>
26.
27.                 <!-- Repository where we can found the storm dependencies -->
28.                 <repository>
29.                         <id>clojars.org</id>
30.                         <url>http://clojars.org/repo</url>
31.                 </repository>
32.
33.         </repositories>
34.
35.         <dependencies>
36.
37.                 <!-- Storm Dependency -->
38.                 <dependency>
39.                         <groupId>storm</groupId>
40.                         <artifactId>storm</artifactId>
41.                         <version>0.9.0.1</version>
42.                         <scope>provided</scope>
43.                 </dependency>
44.                 <dependency>
45.                         <groupId>com.esotericsoftware.kryo</groupId>
46.                         <artifactId>kryo</artifactId>
47.                         <version>2.17</version>
48.                 </dependency>
49.
50.
51.         </dependencies>
52.
53. </project>
```

由于例子中没有输出机制,这里使用写文件的方式来验证程序的正确性。

修改WordCounter类,增加一个局部变量

private FileWriter fileWriter;

修改prepare方法

Java代码

```
1.  @Override
2.  public void prepare(Map stormConf, TopologyContext context) {
3.          this.counters = new HashMap<String, Integer>();
```

```
4.          this.name = context.getThisComponentId();
5.          this.id = context.getThisTaskId();
6.
7.          try {
8.                  this.fileWriter = new FileWriter((String) stormConf.get("outFile"));
9.          } catch (IOException e) {
10.                 throw new RuntimeException("Error write file ["+stormConf.get("outFile")+"]")
    ;
11.         }
12. }
```

修改execute方法

Java代码

```
1.  @Override
2.  public void execute(Tuple input, BasicOutputCollector collector) {
3.          String str = input.getString(0);
4.          /**
5.           * If the word dosn't exist in the map we will create
6.           * this, if not We will add 1
7.           */
8.          if(!counters.containsKey(str)){
9.                  counters.put(str, 1);
10.         }else{
11.                 Integer c = counters.get(str) + 1;
12.                 counters.put(str, c);
13.         }
14.
15.         if(this.fileWriter != null){
16.                 try {
17.                         fileWriter.write("Thread " + Thread.currentThread().getName() + " lo
    g counters==================" + counters);
18.                         fileWriter.write("\r\n");
19.                         fileWriter.write("=================================================
    =");
20.                         fileWriter.write("\r\n");
21.                         fileWriter.flush();
22.                 } catch (IOException e) {
23.                         e.printStackTrace();
24.                 }
25.         }
26. }
```

修改TopologyMain的main方法

Java代码

```
1.  public class TopologyMain {
2.          public static void main(String[] args) throws InterruptedException, AlreadyAliveExcep
    tion, InvalidTopologyException {
3.
4.          //Topology definition
5.                  TopologyBuilder builder = new TopologyBuilder();
6.                  builder.setSpout("word-reader",new WordReader());
7.                  builder.setBolt("word-normalizer", new WordNormalizer())
8.                          .shuffleGrouping("word-reader");
9.                  builder.setBolt("word-counter", new WordCounter(),(Number)1)
10.                         .fieldsGrouping("word-normalizer", new Fields("word"));
11.
12.         //Configuration
13.                 Config conf = new Config();
```

```
                conf.put("wordsFile", args[0]);
                conf.put("outFile", args[1]);
                conf.setDebug(true);
        //Topology run
                conf.put(Config.TOPOLOGY_MAX_SPOUT_PENDING, 1);

            conf.setNumWorkers(3);
            StormSubmitter.submitTopology("Getting-Started-Toplogie", conf, builder.createTopology());

//              LocalCluster cluster = new LocalCluster();
//              cluster.submitTopology("Getting-Started-Toplogie", conf, builder.createTopology());
//              Thread.sleep(1000);
//              cluster.shutdown();

        }
}
```

mvn install 打包之后,得到Getting-Started-0.0.1-SNAPSHOT.jar文件

创建数据源文件 words.txt

Java代码

```
Storm
test
are
great
is
an
Storm
simple
application
but
very
powerful
really
Storm
is
great
```

提交到集群运行

到storm安装目录

bin/storm Getting-Started-0.0.1-SNAPSHOT.jar TopologyMain /tmp/words.txt /tmp/words-result.txt

运行之后,在/tmp 目录下发现新生成的words-result.txt文件,内容如下

Java代码

```
Thread Thread-16-word-counter log counters==================={storm=1}
=====================================================Thread Thread-16-word-counter log counters==================={test=1, storm=1}
=====================================================Thread Thread-16-word-counter log counters==================={are=1, test=1, storm=1}
=====================================================Thread Thread-16-word-counter log counters==================={great=1, are=1, test=1, storm=1}
=====================================================Thread Thread-16-word-counter log counters
```

```
==========={is=1, great=1, are=1, test=1, storm=1}
==========Thread Thread-16-word-counter log counters
==========={is=1, great=1, are=1, test=1, an=1, storm=1}
==========Thread Thread-16-word-counter log counters
==========={is=1, great=1, are=1, test=1, an=1, storm=2}
==========Thread Thread-16-word-counter log counters
==========={is=1, great=1, are=1, test=1, simple=1, an=1, storm=2}
==========Thread Thread-16-word-counter log counters
==========={application=1, is=1, great=1, are=1, test=1, simple=1, an=1, storm=2}
==========Thread Thread-16-word-counter log counters
==========={but=1, application=1, is=1, great=1, are=1, test=1, simple=1, an=1, storm
=2}
==========Thread Thread-16-word-counter log counters
==========={but=1, application=1, is=1, great=1, are=1, test=1, simple=1, an=1, storm
=2, very=1}
==========Thread Thread-16-word-counter log counters
==========={but=1, application=1, is=1, great=1, are=1, test=1, simple=1, an=1, storm
=2, powerful=1, very=1}
==========Thread Thread-16-word-counter log counters
==========={really=1, but=1, application=1, is=1, great=1, are=1, test=1, simple=1, a
n=1, storm=2, powerful=1, very=1}
==========Thread Thread-16-word-counter log counters
==========={really=1, but=1, application=1, is=1, great=1, are=1, test=1, simple=1, a
n=1, storm=3, powerful=1, very=1}
==========Thread Thread-16-word-counter log counters
==========={really=1, but=1, application=1, is=2, great=1, are=1, test=1, simple=1, a
n=1, storm=3, powerful=1, very=1}
==========Thread Thread-16-word-counter log counters
==========={really=1, but=1, application=1, is=2, great=2, are=1, test=1, simple=1, a
n=1, storm=3, powerful=1, very=1}
==========
```

参考:

https://github.com/nathanmarz/storm/wiki/Setting-up-a-Storm-cluster

http://zookeeper.apache.org/doc/r3.3.3/zookeeperStarted.html

http://blog.csdn.net/thermosym/article/details/9254799

storm-book-examples-ch02-getting_started_ex.rar (4.7 KB)
下载次数: 12

分享到:

guava学习笔记 | Storm本地模式环境问题

2014-01-02 12:39 | 浏览 1005 | 评论(0) | 分类:开源软件 | 相关推荐 MORE

评论

发表评论

您还没有登录,请您登录后再发表评论