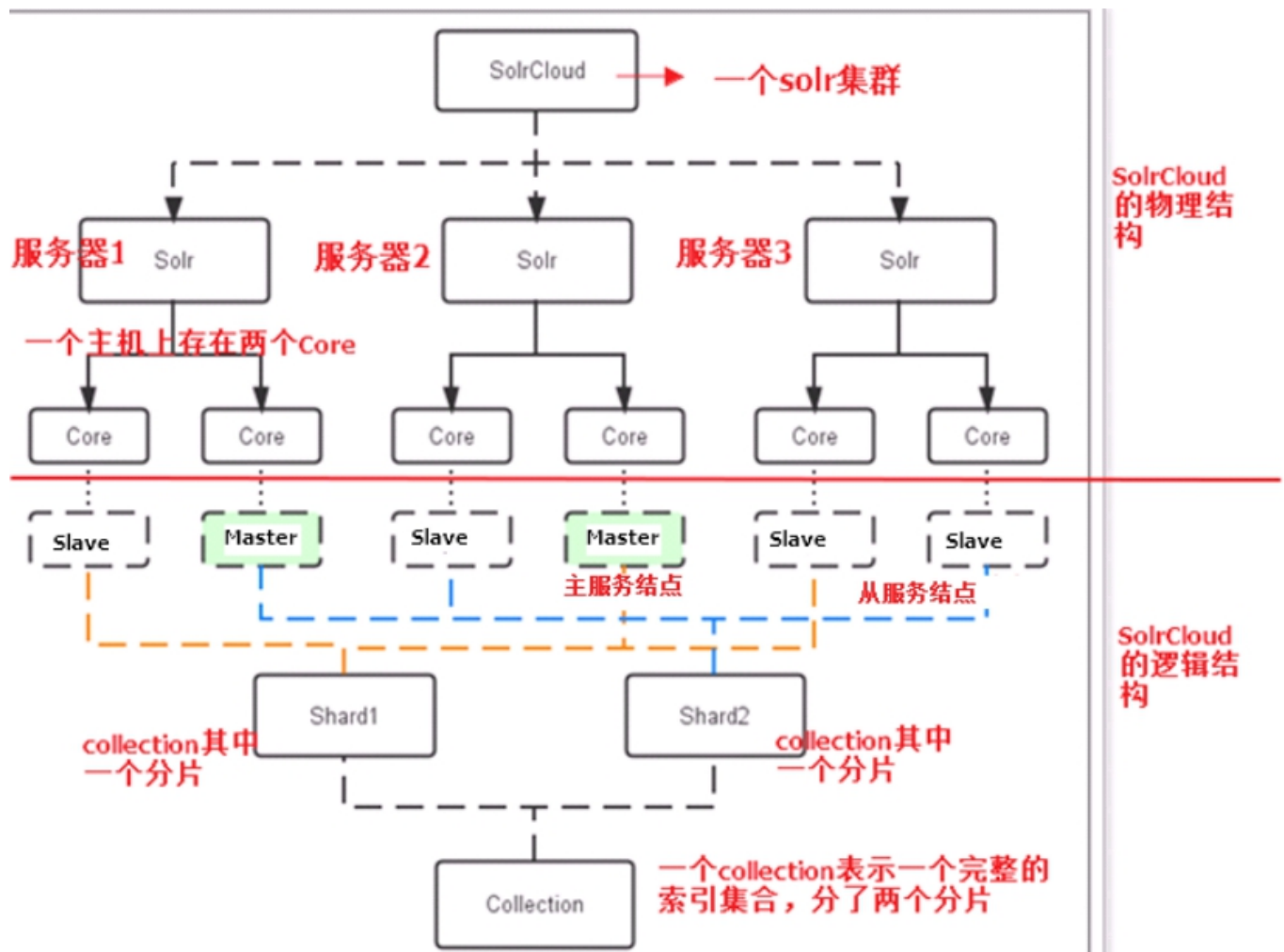


|1 Solr集群

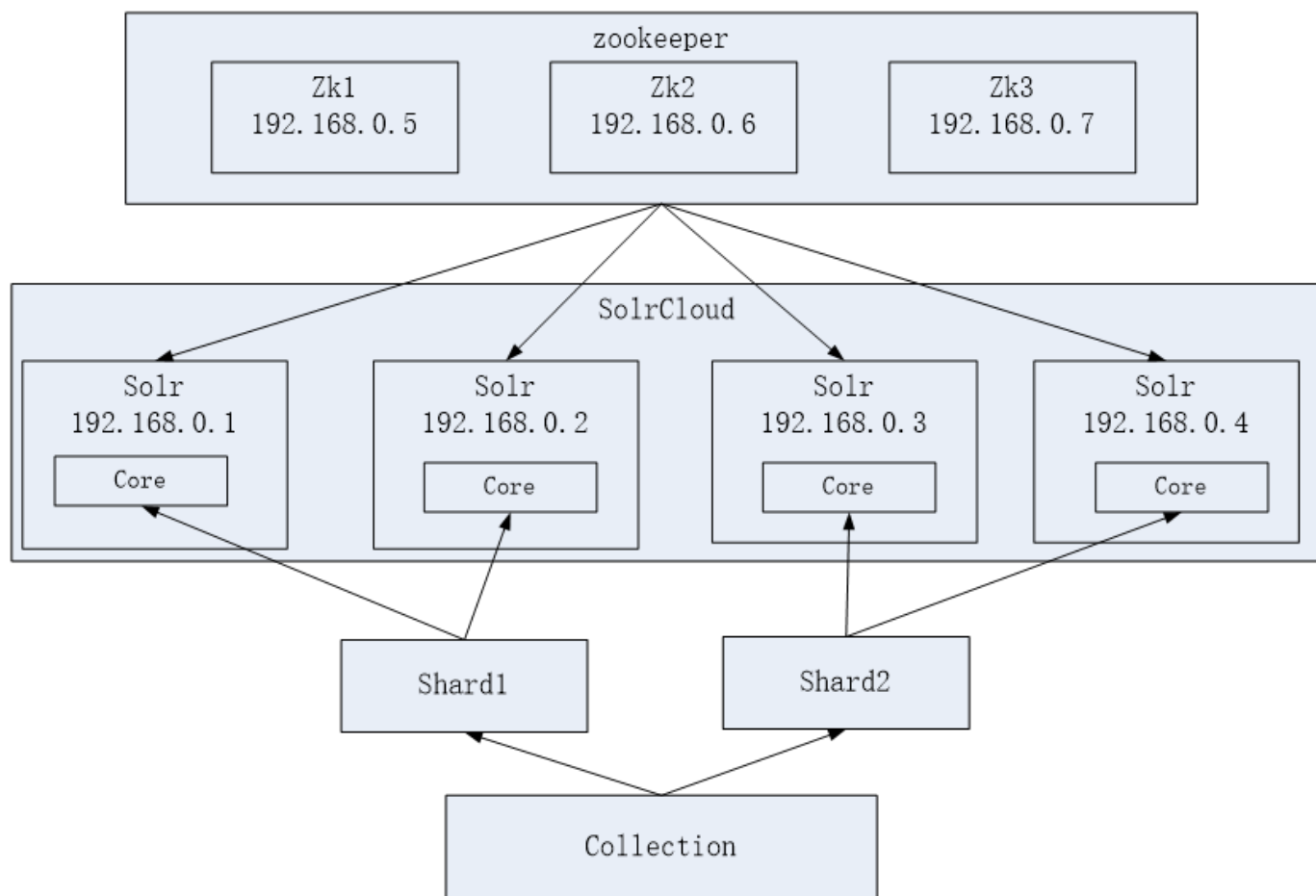
1.1 Solr集群的架构

SolrCloud

需要用到solr+zookeeper



要完成的集群结构



1.2 Zookeeper

1、集群管理

主从的管理、负载均衡、高可用的管理。集群的入口。Zookeeper必须是集群才能保证高可用。Zookeeper有选举和投票的机制。集群中至少应该有三个节点。

2、配置文件的集中管理

搭建solr集群时，需要把Solr的配置文件上传zookeeper，让zookeeper统一管理。每个节点都到zookeeper上取配置文件。

3、分布式锁

4、名字服务

1.3 集群需要的服务器

Zookeeper: 3台

Solr: 4台

伪分布式，zookeeper三个实例、tomcat（solr）需要四个实例。

Zookeeper需要安装jdk。

1.4 集群搭建步骤

第一部分：Zookeeper集群搭建

第一步：需要把zookeeper的安装包上传到服务器。

第二步：把zookeeper解压。

第三步：把zookeeper向/usr/local/solr-cloud目录下复制三份。

第三步：配置zookeeper。

- 1、在zookeeper01目录下创建一个data文件夹。
- 2、在data目录下创建一个myid的文件
- 3、Myid的内容为1（02对应“2”，03对应“3”）
- 4、进入conf文件，把zoo_sample.cfg文件改名为zoo.cfg
- 5、修改zoo.cfg，把dataDir=属性指定为刚创建的data文件夹。
- 6、修改zoo.cfg，把clientPort指定为不冲突的端口号（01:2181、02:2182、03:2183）
- 7、在zoo.cfg中添加如下内容：

server.1=192.168.25.154:2881:3881

server.2=192.168.25.154:2882:3882

server.3=192.168.25.154:2883:3883

(根据自己ip地址进行修改)

```
# do not use /tmp for storage, /tmp here is just
# example sake.
dataDir=/usr/local/solr-cloud/zookeeper01/data/
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.h
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
server.1=192.168.25.154:2881:3881
server.2=192.168.25.154:2882:3882
server.3=192.168.25.154:2883:3883
```

8、Zookeeper02、03都要重复以上步骤。

第四步：启动zookeeper。

Zookeeper的目录下有一个bin目录。使用zkServer.sh启动zookeeper服务。

启动：./zkServer.sh start

关闭: `./zkServer.sh stop`

查看服务状态: `./zkServer.sh status`

第二部分: 搭建solr集群

第一步: 安装四个tomcat, 修改其端口号不能冲突。8080~8083

第二步: 向tomcat下部署solr。把单机版的solr工程复制到tomcat下即可。

第三步: 为每个solr实例创建一solrhome。

第四步: 为每个solr实例关联对应的solrhome。修改web.xml

第五步: 修改每个solrhome下的solr.xml文件。修改host、hostPort两个属性。分别是对应的ip及端口号。

```
<solrcloud>
  <str name="host">${host:192.168.25.154}</str>
  <int name="hostPort">${jetty.port:8080}</int>
  <str name="hostContext">${hostContext:solr}</str>
  <int name="zkClientTimeout">${zkClientTimeout:30000}</int>
  <bool name="genericCoreNodeNames">${genericCoreNodeNames:true}</bool>
</solrcloud>

<dispatcherFactory name="dispatcherFactory">
```

第六步: 把配置文件上传到zookeeper。需要使用

`/root/solr-4.10.3/example/scripts/cloud-scripts/zkcli.sh`命令上传配置文件。

把`/usr/local/solr-cloud/solrhome01/collection1/conf`目录上传到zookeeper。

需要zookeeper集群已经启动。

```
1 ./zkcli.sh -zkhost 192.168.25.154:2181,192.168.25.154:2182,192.168.25.154:2183
  -cmd upconfig -confdir /usr/local/solr-cloud/solrhome01/collection1/conf -
  confname myconf
```

第七步: 查看是否上传成功。

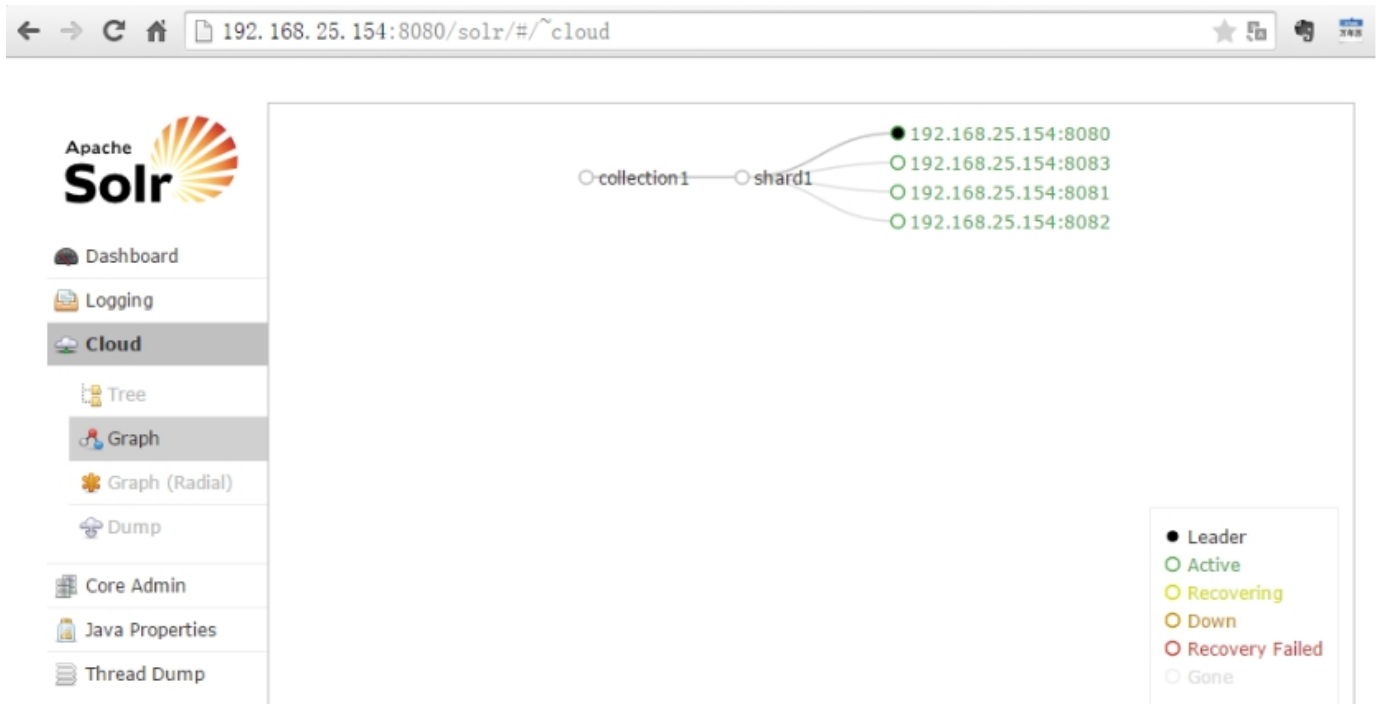
使用zookeeper的`zkCli.sh`命令。

第八步: 告诉solr实例zookeeper的位置。需要修改tomcat的`catalina.sh`添加

`JAVA_OPTS="-DzkHost=192.168.25.154:2181,192.168.25.154:2182,192.168.25.154:2183"` (添加在251行)

每个节点都需要添加。

第九步: 启动每个solr实例。



第十步：集群分片。

将集群分为两片，每片两个副本。

<http://192.168.25.154:8080/solr/admin/collections?>

[action=CREATE&name=collection2&numShards=2&replicationFactor=2](http://192.168.25.154:8080/solr/admin/collections?action=CREATE&name=collection2&numShards=2&replicationFactor=2)

获得如下显示表明成功

```
1 This XML file does not appear to have any style information associated with
  it. The document tree is shown below.
2 <response>
3   <lst name="responseHeader">
4     <int name="status">0</int>
5     <int name="QTime">10514</int>
6   </lst>
7   <lst name="success">
8     <lst>
9       <lst name="responseHeader">
10        <int name="status">0</int>
11        <int name="QTime">9264</int>
12      </lst>
13      <str name="core">collection2_shard1_replica1</str>
14    </lst>
15  </lst>
16  <lst name="responseHeader">
17    <int name="status">0</int>
18    <int name="QTime">9436</int>
19  </lst>
20  <str name="core">collection2_shard2_replica2</str>
```

```

21 </lst>
22 <lst>
23 <lst name="responseHeader">
24 <int name="status">0</int>
25 <int name="QTime">9637</int>
26 </lst>
27 <str name="core">collection2_shard2_replica1</str>
28 </lst>
29 <lst>
30 <lst name="responseHeader">
31 <int name="status">0</int>
32 <int name="QTime">9814</int>
33 </lst>
34 <str name="core">collection2_shard1_replica2</str>
35 </lst>
36 </lst>
37 </response>

```



第十一步：删除不用collection1

<http://192.168.25.154:8080/solr/admin/collections?action=DELETE&name=collection1>

获得如下显示表明成功

```

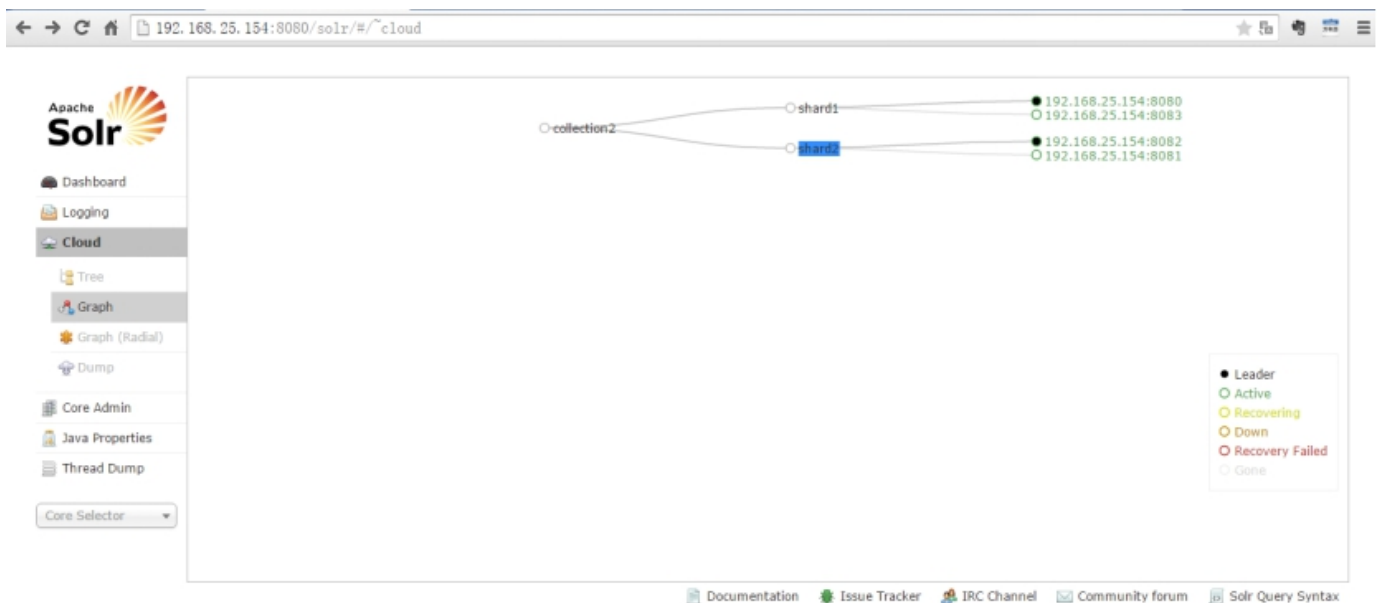
1 This XML file does not appear to have any style information associated with
  it. The document tree is shown below.
2 <response>
3 <lst name="responseHeader">
4 <int name="status">0</int>
5 <int name="QTime">1106</int>

```

```

6 </lst>
7 <lst name="success">
8 <lst name="192.168.179.128:8081_solr">
9 <lst name="responseHeader">
10 <int name="status">0</int>
11 <int name="QTime">467</int>
12 </lst>
13 </lst>
14 <lst name="192.168.179.128:8080_solr">
15 <lst name="responseHeader">
16 <int name="status">0</int>
17 <int name="QTime">501</int>
18 </lst>
19 </lst>
20 <lst name="192.168.179.128:8083_solr">
21 <lst name="responseHeader">
22 <int name="status">0</int>
23 <int name="QTime">535</int>
24 </lst>
25 </lst>
26 <lst name="192.168.179.128:8082_solr">
27 <lst name="responseHeader">
28 <int name="status">0</int>
29 <int name="QTime">594</int>
30 </lst>
31 </lst>
32 </lst>
33 </response>

```



1.5 使用solrJ连接集群

```
1  @Test
2  public void testSolrClout() throws Exception {
3      //创建一个SolrServer对象
4      CloudSolrServer solrServer = new
        CloudSolrServer("192.168.25.154:2181,192.168.25.154:2182,192.168.25.154:2183")
        ;
5      //设置默认的collection
6      solrServer.setDefaultCollection("collection2");
7      //创建一个文档对象
8      SolrInputDocument document = new SolrInputDocument();
9      document.addField("id", "test01");
10     document.addField("item_title", "title1");
11     //添加文档
12     solrServer.add(document);
13     //提交
14     solrServer.commit();
15 }
```