Atitit.获取某个服务 网络邻居列表 解决方案

# 原理，带入某个ip扫描从0---255

很快，多线程几秒就可以出来。

使用CountDownLatch来join线程..

返回 [{

"ip":"192.168.2.114",

"url":"http://@ip@:8080/cms/list\_detail.html"

}]

作者::  ★(attilax)>>>  绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

package com.attilax.net;

import java.net.InetAddress;

import java.net.UnknownHostException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.concurrent.CountDownLatch;

import javax.management.RuntimeErrorException;

import com.attilax.core;

import com.attilax.json.AtiJson;

import com.attilax.lang.text.strUtil;

import com.csmy.my.center.util.zto.HttpUtil;

public class IPUtil {

public List<Map> urls=new ArrayList<Map>();

public static void main(String[] args) throws UnknownHostException {

// System.out.println( getIp());

String url = "http://@ip@:8080/cms/list\_detail.html";

IPUtil ipx=new IPUtil();

List<Map> urls2 = ipx. getActIp(url);

System.out.println( AtiJson.toJson(urls2));

////////

}

public List<Map> getActIp(final String url) {

// String a = subnet\_url("http://192.168.0.111/lime/aa.jsp");

// System.out.println(a);

// 192.168.2.114

String ip;

try {

ip = InetAddress.getLocalHost().getHostAddress();

} catch (UnknownHostException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

throw new RuntimeException(e1);

}

System.out.println(ip);

String subnet = subnet(ip);

System.out.println(subnet);

//String okIp = "";

CountDownLatch end = new CountDownLatch(254);

for (int i = 1; i < 255; i++) {

final int tmp\_i=i;

Runnable runnable = new Runnable() {

@Override

public void run() {

try {

String ip\_item = subnet + "." + String.valueOf(tmp\_i);

String url2 = url.replace("@ip@", ip\_item);

System.out.println(url2);

HttpUtil.sendGet(url2);

// okIp = url;

Map m=new HashMap();

m.put("ip", ip\_item);

m.put("url", url);

urls.add(m);

} catch (Exception e) {

// e.printStackTrace();

String ip\_item = subnet + "." + String.valueOf(tmp\_i);

System.out.println( ip\_item+ "err:"+e.getMessage() );

}

end.countDown();

}

};

core.execMeth\_Ays(runnable, "threadName"+String.valueOf(i));

}

try{

end.await(); //等待end状态变为0，即为比赛结束

}catch (InterruptedException e) {

throw new RuntimeException(e);

}finally{

System.out.println("Race ends!");

}

// exe.shutdown();

return this.urls;

}

/\*\*

\* ret subnet from url

\* @param url

\* @return

\*/

public static String subnet\_url(String url) {

String[] a=strUtil.splitByMultiChar(url, "//,/");

String host\_port=a[1];

return subnet(host\_port);

}

public static String subnet(String host\_port) {

int lastIdx=host\_port.lastIndexOf(".");

return host\_port.substring(0,lastIdx);

}

public static boolean isInternalIp(String ipAddress){

boolean isInnerIp = false;

long ipNum = getIpNum(ipAddress);

/\*\*

私有IP：

A类 10.0.0.0-10.255.255.255

B类 172.16.0.0-172.31.255.255

C类 192.168.0.0-192.168.255.255

\*\*/

long aBegin = getIpNum("10.0.0.0");

long aEnd = getIpNum("10.255.255.255");

long bBegin = getIpNum("172.16.0.0");

long bEnd = getIpNum("172.31.255.255");

long cBegin = getIpNum("192.168.0.0");

long cEnd = getIpNum("192.168.255.255");

isInnerIp = isInnerIp(ipNum,aBegin,aEnd) || isInnerIp(ipNum,bBegin,bEnd) || isInnerIp(ipNum,cBegin,cEnd) || ipAddress.equals("127.0.0.1"); //访问本地localhost获取为127.0.0.1

return isInnerIp;

}

private static long getIpNum(String ipAddress) {

String [] ip = ipAddress.split("\\.");

long a = Integer.parseInt(ip[0]);

long b = Integer.parseInt(ip[1]);

long c = Integer.parseInt(ip[2]);

long d = Integer.parseInt(ip[3]);

long ipNum = a \* 256 \* 256 \* 256 + b \* 256 \* 256 + c \* 256 + d;

return ipNum;

}

private static boolean isInnerIp(long userIp,long begin,long end){

return (userIp>=begin) && (userIp<=end);

}

public static String getIp()

{

InetAddress ia = null;

try {

ia = InetAddress.getLocalHost();

} catch (UnknownHostException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// System.out.println();

return ia.getHostAddress();

}

public static String subnet()

{

String ip= getIp();

int lastIdx=ip.lastIndexOf(".");

return ip.substring(0,lastIdx);

}

}