**Zookeeper应用：服务端上下线**

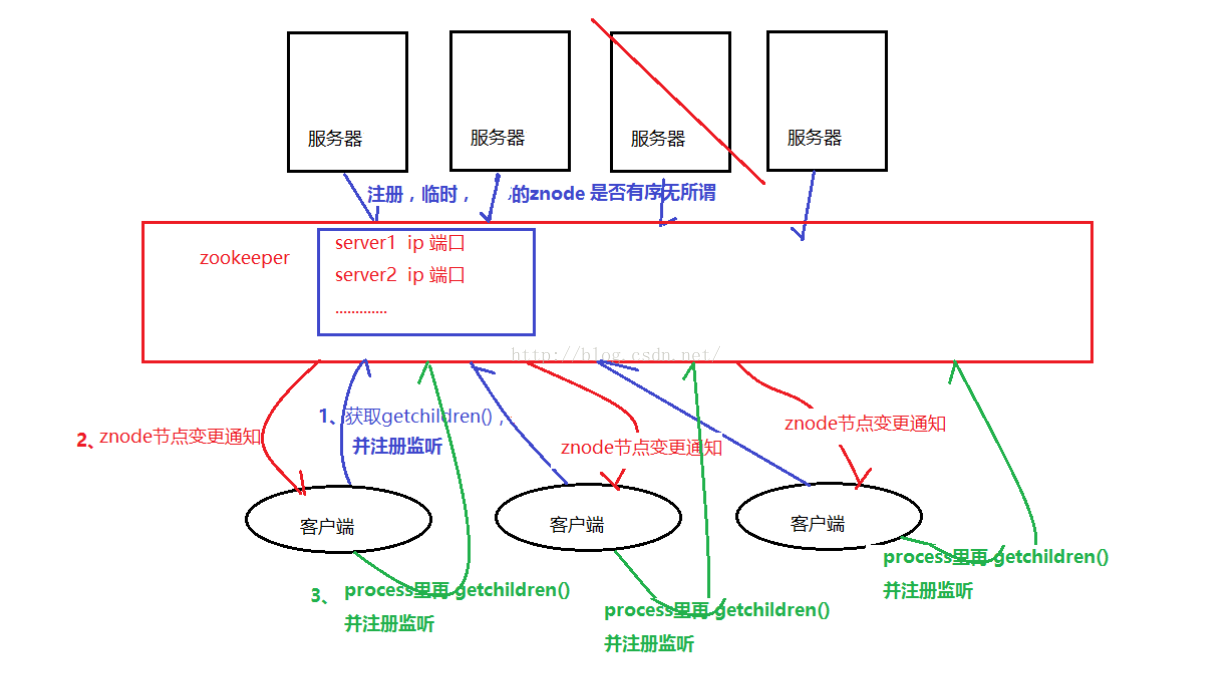
原创 2016年11月17日 13:20:22

* 341

需求

客户端感知服务器的上下线。

示意图



步骤

服务端：

1、所有机子向Zookeeper注册，注册znode为临时的。

2、有机子下线，连接断开后被Zookeeper自动删除，触发监听事件。

3、有机子上线，触发监听事件。

客户端：

1、连接Zookeeper，获取服务器注册的znode，getchildren()，并注册监听。

2、当Zookeeper触发监听，会rpc远程调用process。

3、process调用getchildren().

服务端代码

**[java]** [view plain](http://blog.csdn.net/zengmingen/article/details/53186595) [copy](http://blog.csdn.net/zengmingen/article/details/53186595)

1. **package** hello.zookeeper.schange;
3. **import** org.apache.zookeeper.CreateMode;
4. **import** org.apache.zookeeper.WatchedEvent;
5. **import** org.apache.zookeeper.Watcher;
6. **import** org.apache.zookeeper.ZooDefs.Ids;
7. **import** org.apache.zookeeper.ZooKeeper;
9. **public** **class** Server {
11. **private** String connUrl = "zk1:2181,zk2:2181,zk3:2181";
12. **private** **int** outTime = 1000;
13. **private** String parentNode="/servers/";//Zookeeper先建好这个持久节点
15. **private** ZooKeeper zk = **null**;
17. **public** **void** getConnection() **throws** Exception {
19. zk = **new** ZooKeeper(connUrl, outTime, **new** Watcher() {
21. @Override
22. **public** **void** process(WatchedEvent event) {
23. System.out.println("Server Watcher：" + event.getType() + "---" + event.getPath());
24. }
25. });
26. }
28. **public** **void** registerServer(String server) **throws** Exception {
29. String msg = zk.create(parentNode+server, server.getBytes(), Ids.OPEN\_ACL\_UNSAFE, CreateMode.EPHEMERAL\_SEQUENTIAL);
30. System.out.println("registerServer：" + msg);
31. }
33. **public** **static** **void** main(String[] args) **throws** Exception {
35. String name = "server4";
36. Server server = **new** Server();
37. server.getConnection();
38. server.registerServer(name);
40. System.out.println(name + "----start------");
42. Thread.sleep(Long.MAX\_VALUE);
43. }
45. }

客户端代码

**[java]** [view plain](http://blog.csdn.net/zengmingen/article/details/53186595) [copy](http://blog.csdn.net/zengmingen/article/details/53186595)

1. **package** hello.zookeeper.schange;
3. **import** java.util.List;
5. **import** org.apache.zookeeper.WatchedEvent;
6. **import** org.apache.zookeeper.Watcher;
7. **import** org.apache.zookeeper.ZooKeeper;
9. **public** **class** Client {
11. **private** String connUrl = "zk1:2181,zk2:2181,zk3:2181";
12. **private** **int** outTime = 1000;
13. **private** String parentNode="/servers";
15. **private** ZooKeeper zk = **null**;
17. **public** **void** getConnection() **throws** Exception {
19. zk = **new** ZooKeeper(connUrl, outTime, **new** Watcher() {
21. @Override
22. **public** **void** process(WatchedEvent event) {
23. System.out.println("Client Watcher：" + event.getType() + "---" + event.getPath());
24. **try** {
25. getServerList();
26. } **catch** (Exception e) {
27. e.printStackTrace();
28. }
29. }
30. });
31. }

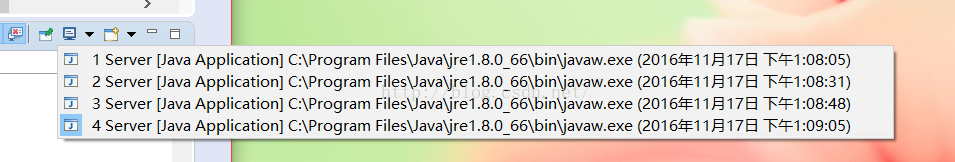
34. **public** **void** getServerList() **throws** Exception{
36. List<String> children = zk.getChildren(parentNode, **true**);
37. **if**(children!=**null**&&children.size()>0){
38. System.out.println("服务器列表如下：");
39. **for**(String child:children){
40. System.out.println(child);
41. }
42. }
43. }

46. **public** **static** **void** main(String[] args) **throws** Exception {
47. Client client=**new** Client();
48. client.getConnection();
49. client.getServerList();
51. Thread.sleep(Long.MAX\_VALUE);
52. }

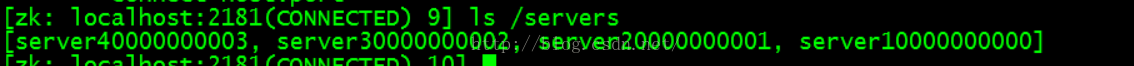
55. }

运行结果

开了4台服务端，3台客户端



Zookeeper里/servers节点下有4台注册在线的服务器



客户端运行获取服务端列表



服务器下线一台，三个客户端都受到通知



-------------

更多的[**Java**](http://lib.csdn.net/base/java)，Angular，[**Android**](http://lib.csdn.net/base/android)，[**大数据**](http://lib.csdn.net/base/hadoop)，J2EE，[**Python**](http://lib.csdn.net/base/python)，[**数据库**](http://lib.csdn.net/base/mysql)，[**Linux**](http://lib.csdn.net/base/linux)，Java[**架构**](http://lib.csdn.net/base/architecture)师，：

<http://www.cnblogs.com/zengmiaogen/p/7083694.html>