# ZooKeeper提供了什么

## 文件系统

Zookeeper的文件系统是znode组成，可以看做是一个文件树。

Znode节点类型

* PERSISTENT-持久化目录节点
* PERSISTENT\_SEQUENTIAL-持久化顺序编号目录节点
* EPHEMERAL-临时目录节点
* EPHEMERAL\_SEQUENTIAL-临时顺序编号目录节点

临时节点是指在与服务器断开连接后，就会zookeeper服务自动被删除的节点。而持久性节点即便在与服务器断开连接后，zookeeper服务也不会清理。

顺序节点可以在创建节点时传入相同的path，如/root/test,在创建时会自动给test进行编号。

而非顺序节点不能创建两个相同path的节点，不然会报异常

## Watcher通知机制

|  |  |  |
| --- | --- | --- |
|  | event For “/path” | event For “/path/child” |
| create(“/path”) | EventType.NodeCreated | 无 |
| delete(“/path”) | EventType.NodeDeleted | 无 |
| setData(“/path”) | EventType.NodeDataChanged | 无 |
| create(“/path/child”) | EventType.NodeChildrenChanged（getChild） | EventType.NodeCreated |
| delete(“/path/child”) | EventType.NodeChildrenChanged（getChild） | EventType.NodeDeleted |
| setData(“/path/child”) | 无 | EventType.NodeDataChanged |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| event For “/path” | Default Watcher | exists(“/path”) | getData(“/path”) | getChildren(“/path”) |
| EventType.None | √ | √ | √ | √ |
| EventType.NodeCreated |  | √ | √ |  |
| EventType.NodeDeleted |  | √ | √ |  |
| EventType.NodeDataChanged |  | √ | √ |  |
| EventType.NodeChildrenChanged |  |  |  | √ |

New ZooKeeper时注册的watcher叫default watcher，它不是一次性的，只对client的连接状态变化作出反应,而其他watcher都是一次性的。

比如说，在调用exists(“/path”)之后，当path的数据发生变化时，会触发EventType.NodeDataChange事件，但是数据再次变化时，事件将不会再次触发。

# Zookeeper可以做什么

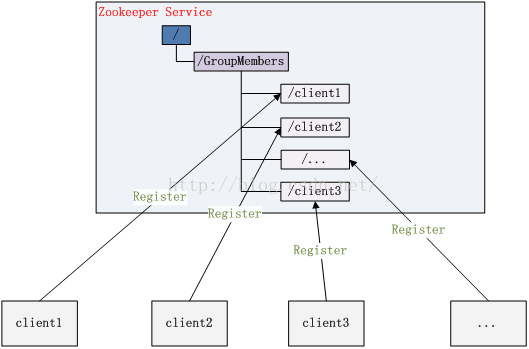
Zookeeper的功能都是通过他文件系统和监听机制来实现的。

## 集群管理

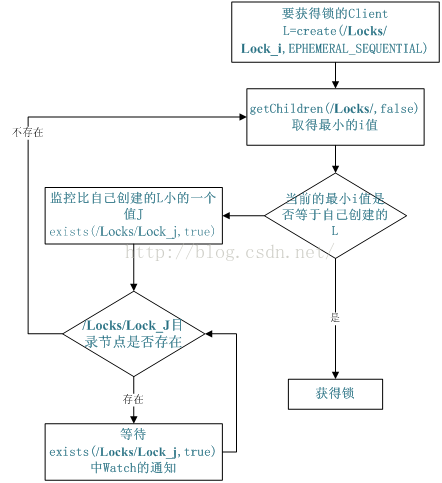
所谓集群管理无在乎两点：是否有机器退出和加入、选举master。

对于第一点，所有机器约定在父目录GroupMembers下创建临时目录节点，然后监听父目录节点的子节点变化消息。一旦有机器挂掉，该机器与 zookeeper的连接断开，其所创建的临时目录节点被删除，所有其他机器都收到通知：某个兄弟目录被删除，于是，所有人都知道：它上船了。

新机器加入也是类似，所有机器收到通知：新兄弟目录加入，highcount又有了，对于第二点，我们稍微改变一下，所有机器创建临时顺序编号目录节点，每次选取编号最小的机器作为master就好。



## 分布式锁



# Zookeeper客户端

Zookeeper.NET

## Zookeeper客户端连接

AutoResetEvent autoResetEvent = new AutoResetEvent(false);

Watcher watcher = new Watcher(autoResetEvent);

ZooKeeper zk = new ZooKeeper(Zookeeper.Host, new TimeSpan(0, 0, 0, Zookeeper.SessionTimeout), watcher);

bool waitzk = autoResetEvent.WaitOne(15000);

if (!waitzk)

{

throw new Exception("区域{0}的ZooKeeper服务连接超时");

}

连接是用的异步连接，所以在new一个zookeeper后，并不一定已经连接上服务，这时候调用create、exists等方法就会出现异常，这里使用AutoResetEvent来调节。当连接上服务时，会触发watcher监听。