**基于ZooKeeper的Dubbo注册中心**

**Zookeeper注册中心安装**

建议使用dubbo-2.3.3以上版本的zookeeper注册中心客户端。[Zookeeper](http://zookeeper.apache.org/)是Apache Hadoop的子项目，强度相对较好，建议生产环境使用该注册中心。Dubbo未对Zookeeper服务器端做任何侵入修改，只需安装原生的Zookeeper服务器即可，所有注册中心逻辑适配都在调用Zookeeper客户端时完成。如果需要，可以考虑使用taobao的zookeeper监控：<http://rdc.taobao.com/team/jm/archives/1450>

**安装:**

wget [http://www.apache.org/dist//zookeeper/zookeeper-3.3.3/zookeeper-3.3.3.tar.gz](http://www.apache.org/dist/zookeeper/zookeeper-3.3.3/zookeeper-3.3.3.tar.gz)

tar zxvf zookeeper-3.3.3.tar.gz

cd zookeeper-3.3.3

cp conf/zoo\_sample.cfg conf/zoo.cfg

**配置:**

|  |
| --- |
| vi conf/zoo.cfg |

如果不需要集群，zoo.cfg的内容如下：(其中data目录需改成你真实输出目录)

|  |
| --- |
| tickTime=2000  initLimit=10  syncLimit=5  dataDir=/home/dubbo/zookeeper-3.3.3/data  clientPort=2181 |

如果需要集群，zoo.cfg的内容如下：(其中data目录和server地址需改成你真实部署机器的信息)

|  |
| --- |
| tickTime=2000  initLimit=10  syncLimit=5  dataDir=/home/dubbo/zookeeper-3.3.3/data  clientPort=2181  server.1=10.20.153.10:2555:3555  server.2=10.20.153.11:2555:3555 |

并在data目录下放置myid文件：(上面zoo.cfg中的dataDir)

|  |
| --- |
| mkdir data  vi myid |

myid指明自己的id，对应上面zoo.cfg中server.后的数字，第一台的内容为1，第二台的内容为2，内容如下：

1

**启动:**

|  |
| --- |
| ./bin/zkServer.sh start |

**停止:**

./bin/zkServer.sh stop

**命令行:** (See: <http://zookeeper.apache.org/doc/r3.3.3/zookeeperAdmin.html>)

telnet 127.0.0.1 2181  
dump

或者

echo dump | nc 127.0.0.1 2181

**用法：**

|  |
| --- |
| dubbo.registry.address=zookeeper://10.20.153.10:2181?backup=10.20.153.11:2181 |

或者

|  |
| --- |
| <dubbo:registryprotocol="zookeeper"address="10.20.153.10:2181,10.20.153.11:2181"/> |

**管理控制台安装**

管理控制台为内部裁剪版本，开源部分主要包含：路由规则，动态配置，服务降级，访问控制，权重调整，负载均衡，等管理功能。

**安装:**

|  |
| --- |
| wget <http://apache.etoak.com/tomcat/tomcat-6/v6.0.35/bin/apache-tomcat-6.0.35.tar.gz>  tar zxvf apache-tomcat-6.0.35.tar.gz  cd apache-tomcat-6.0.35  rm -rf webapps/ROOT |

wget <http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-admin/2.4.1/dubbo-admin-2.4.1.war>

unzip dubbo-admin-2.4.1.war -d webapps/ROOT

**配置:** (或将dubbo.properties放在当前用户目录下)

|  |
| --- |
| vi webapps/ROOT/WEB-INF/dubbo.properties |
| dubbo.registry.address=zookeeper://127.0.0.1:2181  dubbo.admin.root.password=root  dubbo.admin.guest.password=guest |

**启动:**

|  |
| --- |
| ./bin/startup.sh |

**停止:**

./bin/shutdown.sh

**访问:** (用户:root,密码:root 或 用户:guest,密码:guest)

|  |
| --- |
| <http://127.0.0.1:8080/> |

**注册中心抽象**

Dubbo的将注册中心进行抽象，是得它可以外接不同的存储媒介给注册中心提供服务，有ZooKeeper，Memcached，Redis等。Dubbo抽象后，用户可以进行扩展，我们通过分析ZooKeeper这个实现来了解注册中心的低层。进过抽象之后，用户 只需要实现对应的Registry和RegistryFactory就可以了，ZooKeeper就是实现了ZookeeperRegistry，和ZookeeperRegistryFactory。

ZookeeperRegistryFactory的实现很简单，就是返回一个ZookeeperRegistry实例，所以主要的东西是在ZookeeperRegistry中实现的，在ZookeeperRegistry用户需要实现注册URL，注销URL，URL订阅，URL注销订阅和URL查询，在这里面设计到Zookeeper服务端的调用，都被封装到ZookeeperClient中，ZookeeperClient服务进行Server连接，断链；资源的CRUD。

**ZooKeeper的价值**

由于引入了ZooKeeper作为存储媒介，也就把ZooKeeper的特性引进来。首先是负载均衡，单注册中心的承载能力是有限的，在流量达到一定程度的时候就需要分流，负载均衡就是为了分流而存在的，一个ZooKeeper群配合相应的Web应用就可以很容易达到负载均衡；资源同步，单单有负载均衡还不够，节点之间的数据和资源需要同步，ZooKeeper集群就天然具备有这样的功能；命名服务，将树状结构用于维护全局的服务地址列表，服务提供者在启动的时候，向ZK上的指定节点/dubbo/${serviceName}/providers目录下写入自己的URL地址，这个操作就完成了服务的发布。其他特性还有Mast选举，分布式锁等。

附带参考资料：

最佳实践http://rdc.taobao.com/team/jm/archives/1232

功能概述http://www.ibm.com/developerworks/cn/opensource/os-cn-zookeeper/