DUBBO技术文件

技术文件名称：

技术文件编号：

版本：V1.0

|  |  |
| --- | --- |
| 作者： |  |
| 产品名称： |  |
| 项目编码： |  |

**版权声明**

本文中的所有信息均为水鱼有限责任公司秘密信息，务请妥善保管，未经公司明确作出的书面许可，不得为任何目的、以任何形式或手段（包括电子、机械、复印、录音或其他形式）对本文档的任何部分进行复制、存储、引入检索系统或者传播。

水鱼有限责任公司

IT部

# 版本变更记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 拟制/修改日期 | 拟制/修改人 | 修改描述 | 批准人 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录

[版本变更记录 1](#_Toc477884385)

[目录 3](#_Toc477884386)

[1 总体说明 4](#_Toc477884387)

[1.1 编写目的 4](#_Toc477884388)

[1.2 项目背景 4](#_Toc477884389)

[1.3 名词定义 4](#_Toc477884390)

[2 部署结构说明 5](#_Toc477884391)

[2.1 部署图 5](#_Toc477884392)

[2.1.1 组件说明 5](#_Toc477884393)

[2.2 部署说明 6](#_Toc477884394)

[2.2.1 Zookeeperu部署说明： 6](#_Toc477884395)

[2.2.2 Dubbo监听部署说明 : 8](#_Toc477884396)

[2.2.3 Dubbo管理控制部署说明： 10](#_Toc477884397)

[2.3 Dubbo服务发布和使用 11](#_Toc477884398)

[2.4 Dubbo服务消费端调用 12](#_Toc477884399)

# 总体说明

## 编写目的

本技术文档编写目是方便所有开发人员，技术架构人员，能够快速方便的了解dubbo在公司项目体系中的作用。以及能够快速的使用dubbo微服务技术进行开发。

## 项目背景

由于公司目前的所有系统都提供独立的接口，接口方式都有不同，都是通过当前系统直接调用其它系统。随着系统越来越多。关系也错综复杂。所有的接口调用都无法监控。Dubbo接口服务就是为了解决当前问题。

## 名词定义

**服务定义：**服务是围绕服务提供方和服务消费方的，服务提供方实现服务，而服务消费方调用服务。

**服务注册：**通过将服务统一管理起来，可以有效地优化内部应用对服务发布/使用的流程和管理。服务注册中心可以通过特定协议来完成服务对外的统一。Dubbo提供的注册中心提供多个注册方式，当前采用Zookeeper注册中心。

**服务监控：**无论是服务提供方，还是服务消费方，他们都需要对服务调用的实际状态进行有效的监控，从而改进服务质量。

**远程通信协议：**远程通信需要指定通信双方所约定的协议，在保证通信双方理解协议语义的基础上，还要保证高效、稳定的消息传输。公司主要采用dubbo协议和http协议REST风格。

# 部署结构说明

## 部署图



### 组件说明

* 业务数据服务：业务平台数据接口服务，提供各种业务支持。
* Zookeeper服务：用于dubbo接口服务注册功能。
* Dubbo服务:实时监控dubbo运行状况和对dubbo管理控制。
* 健康档案服务:用户的健康档案数据服务，提供查询，录入等。
* 微信服务：给用户提供微信端入口，如预约等。

## 部署说明

### Zookeeperu部署说明：

Zookeeper的安装和配置十分简单, 既可以配置成单机模式, 也可以配置成集群模式. 下面将分别进行介绍.

#### 单机模式

点击这里下载zookeeper的安装包之后, 解压到合适目录. 进入zookeeper目录下的conf子目录, 创建zoo.cfg:

tickTime=2000

dataDir=/Users/apple/zookeeper/data

dataLogDir=/Users/apple/zookeeper/logs

clientPort=4180

tickTime=2000

dataDir=/Users/apple/zookeeper/data

dataLogDir=/Users/apple/zookeeper/logs

clientPort=4180

参数说明:

tickTime: zookeeper中使用的基本时间单位, 毫秒值.

dataDir: 数据目录. 可以是任意目录.

dataLogDir: log目录, 同样可以是任意目录. 如果没有设置该参数, 将使用和dataDir相同的设置.

clientPort: 监听client连接的端口号.

至此, zookeeper的单机模式已经配置好了. 启动server只需运行脚本:

bin/zkServer.sh start

bin/zkServer.sh start

Server启动之后, 就可以启动client连接server了, 执行脚本:

bin/zkCli.sh -server localhost:4180

bin/zkCli.sh -server localhost:4180

#### 伪集群模式

所谓伪集群, 是指在单台机器中启动多个zookeeper进程, 并组成一个集群. 以启动3个zookeeper进程为例.

将zookeeper的目录拷贝2份:

|--zookeeper0

|--zookeeper1

|--zookeeper2

|--zookeeper0

|--zookeeper1

|--zookeeper2

 更改zookeeper0/conf/zoo.cfg文件为:

tickTime=2000

initLimit=5

syncLimit=2

dataDir=/Users/apple/zookeeper0/data

dataLogDir=/Users/apple/zookeeper0/logs

clientPort=4180

server.0=127.0.0.1:8880:7770

server.1=127.0.0.1:8881:7771

server.2=127.0.0.1:8882:7772

tickTime=2000

initLimit=5

syncLimit=2

dataDir=/Users/apple/zookeeper0/data

dataLogDir=/Users/apple/zookeeper0/logs

clientPort=4180

server.0=127.0.0.1:8880:7770

server.1=127.0.0.1:8881:7771

server.2=127.0.0.1:8882:7772

新增了几个参数, 其含义如下:

initLimit: zookeeper集群中的包含多台server, 其中一台为leader, 集群中其余的server为follower. initLimit参数配置初始化连接时, follower和leader之间的最长心跳时间. 此时该参数设置为5, 说明时间限制为5倍tickTime, 即5\*2000=10000ms=10s.

syncLimit: 该参数配置leader和follower之间发送消息, 请求和应答的最大时间长度. 此时该参数设置为2, 说明时间限制为2倍tickTime, 即4000ms.

server.X=A:B:C 其中X是一个数字, 表示这是第几号server. A是该server所在的IP地址. B配置该server和集群中的leader交换消息所使用的端口. C配置选举leader时所使用的端口. 由于配置的是伪集群模式, 所以各个server的B, C参数必须不同.

参照zookeeper0/conf/zoo.cfg, 配置zookeeper1/conf/zoo.cfg, 和zookeeper2/conf/zoo.cfg文件. 只需更改dataDir, dataLogDir, clientPort参数即可.

在之前设置的dataDir中新建myid文件, 写入一个数字, 该数字表示这是第几号server. 该数字必须和zoo.cfg文件中的server.X中的X一一对应.  
/Users/apple/zookeeper0/data/myid文件中写入0, /Users/apple/zookeeper1/data/myid文件中写入1, /Users/apple/zookeeper2/data/myid文件中写入2.

分别进入/Users/apple/zookeeper0/bin, /Users/apple/zookeeper1/bin, /Users/apple/zookeeper2/bin三个目录, 启动server.  
任意选择一个server目录, 启动客户端:

bin/zkCli.sh -server localhost:4180

bin/zkCli.sh -server localhost:4180

#### 集群模式

集群模式的配置和伪集群基本一致.  
由于集群模式下, 各server部署在不同的机器上, 因此各server的conf/zoo.cfg文件可以完全一样.  
下面是一个示例:

tickTime=2000

initLimit=5

syncLimit=2

dataDir=/home/zookeeper/data

dataLogDir=/home/zookeeper/logs

clientPort=4180

server.43=10.1.39.43:2888:3888

server.47=10.1.39.47:2888:3888

server.48=10.1.39.48:2888:3888

tickTime=2000

initLimit=5

syncLimit=2

dataDir=/home/zookeeper/data

dataLogDir=/home/zookeeper/logs

clientPort=4180

server.43=10.1.39.43:2888:3888

server.47=10.1.39.47:2888:3888

server.48=10.1.39.48:2888:3888

示例中部署了3台zookeeper server, 分别部署在10.1.39.43, 10.1.39.47, 10.1.39.48上. 需要注意的是, 各server的dataDir目录下的myid文件中的数字必须不同.

10.1.39.43 server的myid为43, 10.1.39.47 server的myid为47, 10.1.39.48 server的myid为48.

### Dubbo监听部署说明 :

Simple Monitor挂掉不会影响到Consumer和Provider之间的调用，所以用于生产环境不会有风险。

配置好了之后可以结合admin管理后台使用，可以清晰的看到服务的访问记录、成功次数、失败次数等.....

Simple Monitor采用磁盘存储统计信息，请注意安装机器的磁盘限制，如果要集群，建议用[mount](http://alibaba.github.io/dubbo-doc-static/Administrator+Guide-zh.htm)共享磁盘。

 charts目录必须放在jetty.directory下，否则页面上访问不了。

下面就介绍下monitor的配置使用过程：

安装:

wget http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-monitor-simple/2.4.1/dubbo-monitor-simple-2.4.1-assembly.tar.gz

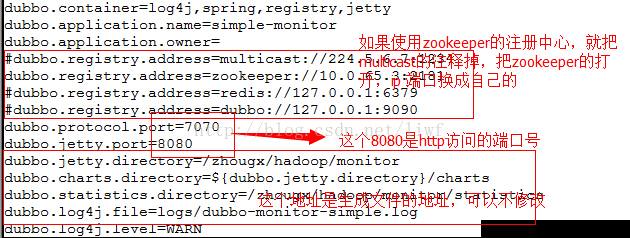
tar zxvf dubbo-monitor-simple-2.4.1-assembly.tar.gz

cd dubbo-monitor-simple-2.4.1

配置:

<span style="font-size:14px;">vi conf/dubbo.properties</span>

编辑包中的这个文件，主要修改一下地方：

  
最主要的就是广播地址：dubbo.registry,address  改成自己的就行了。

还有charts目录就是生成监控图片的额目录，启动后会自动生成这个目录，有点慢可能要稍等一段时间。

集群环境注册中心配置：

dubbo.registry.address=zookeeper://10.135.108.152:2181?backup=10.135.108.153:2181,10.135.108.154:2181

启动:./bin/start.sh

停止:./bin/stop.sh

重启:./bin/restart.sh

调试:./bin/start.sh debug

系统状态:./bin/dump.sh

|  |
| --- |
|  |
|  |

总控入口:./bin/server.sh start

./bin/server.sh stop

./bin/server.sh restart

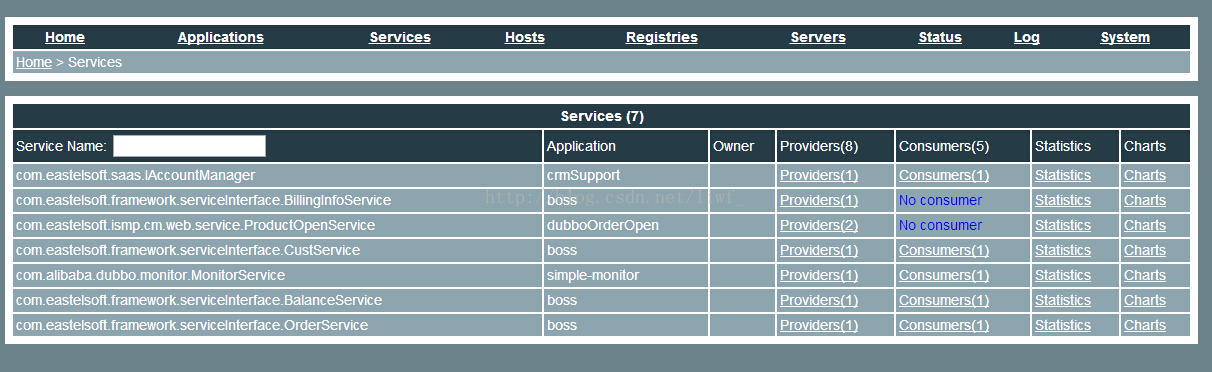
./bin/server.sh debug

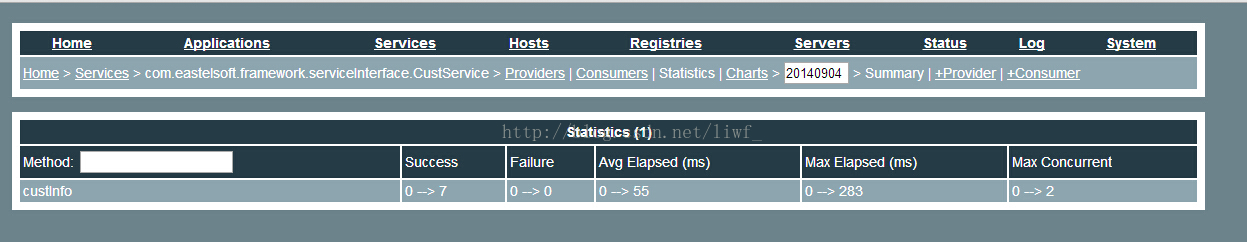
./bin/server.sh dump

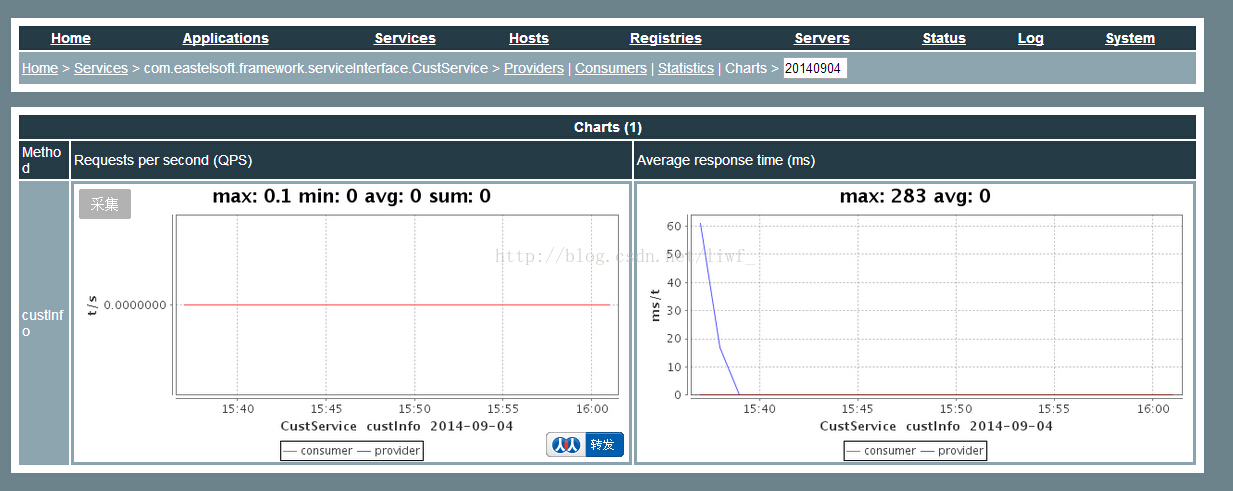
以上都是官方给的启动步骤，按上面的来就行。

启动之后 过一段时候，你在配置文件中指定的目录：dubbo.charts.directory=${dubbo.jetty.directory}/charts  下面就会出来对应的文件，图表目录

然后访问你的界面路径：http://10.0.65.3:8080/   就出现如下的界面了，其他的就看官网的功能介绍吧







**最近有人遇到问题说配置了后还是出不来图片，大致有两种情况：**

1、在配置dubbo.jetty.directory=/aa/bb/monitor 时候 monitor这个目录不存在，dubbo是不会自动给创建这个目录的，他下面的charts和统计的会自动创建；

2、在服务端需要配置监控的配置文件：<dubbo:monitor protocol="registry" />

### Dubbo管理控制部署说明：

部署dubbo-admin到你的tomcat

首先需要将dubbo-admin.war解压后拷贝所有的文件到 webapp下的/ROOT目录中（首先请删除ROOT目录中的所有文件）。 然后在部署目录（即/ROOT目录）下的WEB-INF目录中找到dubbo.properties文件，打开该文件有如下配置项：

dubbo.registry.address=zookeeper://121.40.97.224:2181

dubbo.admin.root.password=root

dubbo.admin.guest.password=guest

其中dubbo.registry.address项需要配置为注册中心的地址和端口，也就是zookeeper的地址和端口

dubbo.admin.root.password为root管理用户的登录密码。

dubbo.admin.guest.password为guest用户的登录密码。

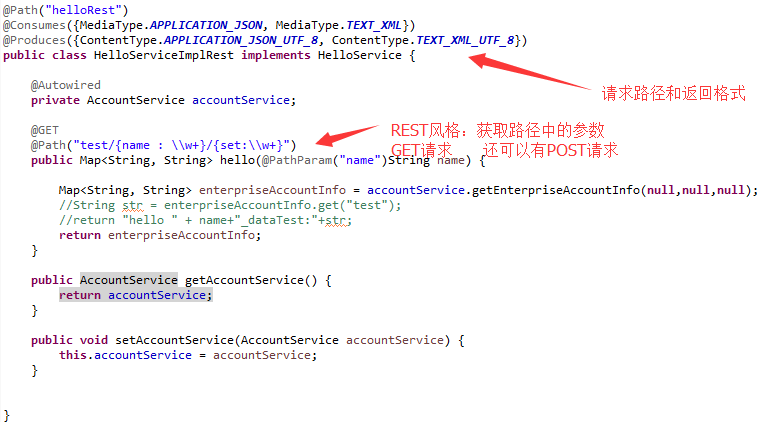
启动tomcat访问dubbo-admin：   
[http://127.0.0.1:8686](http://127.0.0.1:8686/) 可以看到注册中心的系统环境，系统状态已经供者情况

## Dubbo服务发布和使用

1. 编写接口和接口现实类







1. 在Spring配置文件中 配置注册中心：本注册中心是Zookeeper.

<dubbo:registry address="zookeeper://127.0.0.1:2181"/>

1. 在Spring配置文件中 配置通信协议

<!-- use tomcat server -->

<dubbo:protocol name="rest" port="8888" threads="500" contextpath="services" server="tomcat" accepts="500" extension="com.alibaba.dubbo.rpc.protocol.rest.support.LoggingFilter"/>

<dubbo:protocol name="http" port="8889"/>

<dubbo:protocol name="hessian" port="8890"/>

<dubbo:protocol name="webservice" port="8892"/>

1. 在Spring配置文件中 发布dubbo协议服务

<bean id="helloService" class="com.gk.business.bms.interfaces.demo.impl.HelloServiceImpl"/>

<dubbo:service interface="com.gk.business.bms.interfaces.demo.HelloService" ref="helloService" protocol="dubbo"/>

1. 在Spring配置文件中 发布REST风格的HTTP协议服务

<bean id="helloServiceRest" class="com.gk.business.bms.interfaces.demo.impl.HelloServiceImplRest"/>

<dubbo:service interface="com.gk.business.bms.interfaces.demo.HelloService" ref="helloServiceRest" protocol="rest" validation="true"/>

## Dubbo服务消费端调用

1. 在Spring配置文件中配置注册中心：本注册中心是Zookeeper.

<dubbo:registry address="zookeeper://127.0.0.1:2181"/>

1. 设置服务接口

<dubbo:reference interface="com.gk.business.bms.interfaces.demo.HelloService" check="false" id="helloServiceConsumer" protocol="dubbo"></dubbo:reference>

1. 代码调用，可以能过容器加载spring中的接口配置。

