

# zookeeper和dubbo的关系

 daiqing428  2016-04-29  1481 查看

+添加收藏

Dubbo建议使用Zookeeper作为服务的注册中心。

## 1. Zookeeper的作用：

zookeeper用来注册服务和进行负载均衡，哪一个服务由哪一个机器来提供必需让调用者知道，简单来说就是ip地址和服务名称的对应关系。当然也可以通过硬编码的方式把这种对应关系在调用方业务代码中实现，但是如果提供服务的机器挂掉调用者无法知晓，如果不更改代码会继续请求挂掉的机器提供服务。zookeeper通过心跳机制可以检测挂掉的机器并将挂掉机器的ip和服务对应关系从列表中删除。至于支持高并发，简单来说就是横向扩展，在不更改代码的情况通过添加机器来提高运算能力。通过添加新的机器向zookeeper注册服务，服务的提供者多了能服务的客户就多了。

## 2. dubbo：

是管理中间层的工具，在业务层到数据仓库间有非常多服务的接入和服务提供者需要调度，dubbo提供一个框架解决这个问题。

注意这里的dubbo只是一个框架，至于你架子上放什么是完全取决于你的，就像一个汽车骨架，你需要配你的轮子引擎。这个框架中要完成调度必须要有一个分布式的注册中心，储存所有服务的元数据，你可以用zk，也可以用别的，只是大家都用zk。

## 3. zookeeper和dubbo的关系：

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。

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Master选举，分布式锁等。

🔖 标签: [zookeeper](#) [dubbo](#)

## 免费送书活动

- 《智能时代:大数据与智能革命重新定义未来》
- 《C++程序设计语言(第4部分:标准库)(原书第4
- 《计算机科学丛书:Php和Mysql Web开发(原书
- 《数据结构与算法分析:C++语言描述(第2版)》
- 《PHP从入门到精通(第3版)(附光盘)》
- 《计算机科学丛书:Java语言程序设计(基础篇)(
- 《Linux命令行与shell脚本编程大全 第3版》
- 《HTTP权威指南》
- 《细说PHP(第2版)(附DVD光盘1张)》
- 《深入PHP:面向对象、模式与实践(第3版)》
- 《O'Reilly精品图书系列:JavaScript权威指南(第6
- 《吴军博士 浪潮之巅(第3版)(套装共2册)》
- 《图灵程序设计丛书:Python基础教程(第2版)(修
- 《世界著名计算机教材精选:C++面向对象程序
- 《C++ Primer(中文版)(第5版)》
- 《Linux命令行与shell脚本编程大全(第2版)》
- 《深入理解Nginx:模块开发与架构解析(第2版)》
- 《高性能MySQL(第3版)》
- 《PHP开发实战1200例(第1卷)(附DVD-ROM光盘
- 《Android 5.0开发范例代码大全(第4版)》

## 相关文章推荐

- 架构设计：远程调用服务架构设计及zookeeper
- 架构设计：一种远程调用服务的设计构思（zoo
- 15.zookeeper原理解析-客户端与服务器端交互
- 13.zookeeper原理解析-服务器端处理流程之处
- 11.zookeeper原理解析-选举之数据同步
- 9.zookeeper原理解析-选举之QuorumPeerMain
- 7.zookeeper原理解析-序列化之底层通信数据封
- 5.zookeeper原理解析-数据存储之ZKDatabase
- 架构设计：远程调用服务架构设计及zookeeper
- 分布式网站架构后续：zookeeper技术浅析
- 14.zookeeper原理解析-服务器端处理流程之交
- 12.zookeeper原理解析-服务器端处理流程之处
- 10.zookeeper原理解析-选举之选举流程
- 8.zookeeper原理解析-序列化之Record接口
- 6.zookeeper原理解析-数据存储小结
- 4.zookeeper原理解析-数据存储之FileTxnLog&a