## Dubbo是什么

Dubbo是：

一款分布式服务框架

高性能和透明化的RPC远程服务调用方案

SOA服务治理方案

## Dubbo架构



节点角色说明

节点 角色说明

Provider 暴露服务的服务提供方

Consumer 调用远程服务的服务消费方

Registry 服务注册与发现的注册中心

Monitor 统计服务的调用次数和调用时间的监控中心

Container 服务运行容器

调用关系说明

服务容器负责启动，加载，运行服务提供者。

服务提供者在启动时，向注册中心注册自己提供的服务。

服务消费者在启动时，向注册中心订阅自己所需的服务。

注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。

服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。

服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心。

## Dubbo架构具有以下几个特点，分别是连通性、健壮性、伸缩性、以及向未来架构的升级性。

连通性

注册中心负责服务地址的注册与查找，相当于目录服务，服务提供者和消费者只在启动时与注册中心交互，注册中心不转发请求，压力较小

监控中心负责统计各服务调用次数，调用时间等，统计先在内存汇总后每分钟一次发送到监控中心服务器，并以报表展示

服务提供者向注册中心注册其提供的服务，并汇报调用时间到监控中心，此时间不包含网络开销

服务消费者向注册中心获取服务提供者地址列表，并根据负载算法直接调用提供者，同时汇报调用时间到监控中心，此时间包含网络开销

注册中心，服务提供者，服务消费者三者之间均为长连接，监控中心除外

注册中心通过长连接感知服务提供者的存在，服务提供者宕机，注册中心将立即推送事件通知消费者

注册中心和监控中心全部宕机，不影响已运行的提供者和消费者，消费者在本地缓存了提供者列表

注册中心和监控中心都是可选的，服务消费者可以直连服务提供者

健状性

监控中心宕掉不影响使用，只是丢失部分采样数据

数据库宕掉后，注册中心仍能通过缓存提供服务列表查询，但不能注册新服务

注册中心对等集群，任意一台宕掉后，将自动切换到另一台

注册中心全部宕掉后，服务提供者和服务消费者仍能通过本地缓存通讯

服务提供者无状态，任意一台宕掉后，不影响使用

服务提供者全部宕掉后，服务消费者应用将无法使用，并无限次重连等待服务提供者恢复

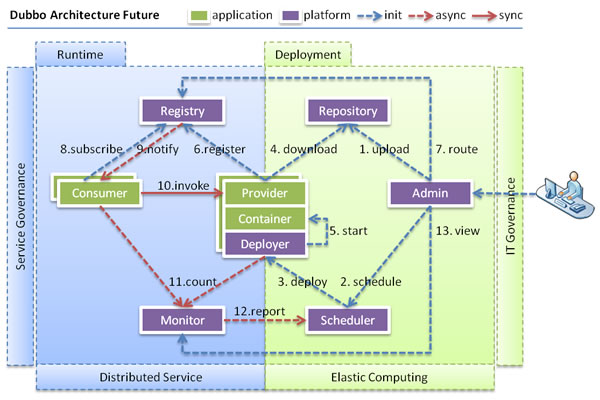
伸缩性

注册中心为对等集群，可动态增加机器部署实例，所有客户端将自动发现新的注册中心

服务提供者无状态，可动态增加机器部署实例，注册中心将推送新的服务提供者信息给消费者

升级性

当服务集群规模进一步扩大，带动IT治理结构进一步升级，需要实现动态部署，进行流动计算，现有分布式服务架构不会带来阻力。下图是未来可能的一种架构：



节点角色说明

| 节点 | 角色说明 |
| --- | --- |
| Deployer | 自动部署服务的本地代理 |
| Repository | 仓库用于存储服务应用发布包 |
| Scheduler | 调度中心基于访问压力自动增减服务提供者 |
| Admin | 统一管理控制台 |
| Registry | 服务注册与发现的注册中心 |
| Monitor | 统计服务的调用次数和调用时间的监控中心 |

## provider.xml 示例

|  |
| --- |
|  |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:dubbo="http://dubbo.apache.org/schema/dubbo"  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd http://dubbo.apache.org/schema/dubbo http://dubbo.apache.org/schema/dubbo/dubbo.xsd">  <dubbo:application name="hello-world-app" />  <dubbo:registry address="multicast://224.5.6.7:1234" />  <dubbo:protocol name="dubbo" port="20880" />  <dubbo:service interface="com.alibaba.dubbo.demo.DemoService" ref="demoServiceLocal" />  <dubbo:reference id="demoServiceRemote" interface="com.alibaba.dubbo.demo.DemoService" />  </beans> |

| 标签 | 用途 | 解释 |
| --- | --- | --- |
| <dubbo:service/> | 服务配置 | 用于暴露一个服务，定义服务的元信息，一个服务可以用多个协议暴露，一个服务也可以注册到多个注册中心 |
| <dubbo:reference/> | 引用配置 | 用于创建一个远程服务代理，一个引用可以指向多个注册中心 |
| <dubbo:protocol/> | 协议配置 | 用于配置提供服务的协议信息，协议由提供方指定，消费方被动接受 |
| <dubbo:application/> | 应用配置 | 用于配置当前应用信息，不管该应用是提供者还是消费者 |
| <dubbo:module/> | 模块配置 | 用于配置当前模块信息，可选 |
| <dubbo:registry/> | 注册中心配置 | 用于配置连接注册中心相关信息 |
| <dubbo:monitor/> | 监控中心配置 | 用于配置连接监控中心相关信息，可选 |
| <dubbo:provider/> | 提供方配置 | 当 ProtocolConfig 和 ServiceConfig 某属性没有配置时，采用此缺省值，可选 |
| <dubbo:consumer/> | 消费方配置 | 当 ReferenceConfig 某属性没有配置时，采用此缺省值，可选 |
| <dubbo:method/> | 方法配置 | 用于 ServiceConfig 和 ReferenceConfig 指定方法级的配置信息 |
| <dubbo:argument/> | 参数配置 | 用于指定方法参数配置 |

# Apache ZooKeeper

## 什么是Apache ZooKeeper？

Apache ZooKeeper是由集群（节点组）使用的一种服务，用于在自身之间协调，并通过稳健的同步技术维护共享数据。ZooKeeper本身是一个分布式应用程序，为写入分布式应用程序提供服务。

ZooKeeper提供的常见服务如下 :

命名服务 - 按名称标识集群中的节点。它类似于DNS，但仅对于节点。

配置管理 - 加入节点的最近的和最新的系统配置信息。

集群管理 - 实时地在集群和节点状态中加入/离开节点。

选举算法 - 选举一个节点作为协调目的的leader。

锁定和同步服务 - 在修改数据的同时锁定数据。此机制可帮助你在连接其他分布式应用程序（如Apache HBase）时进行自动故障恢复。

高度可靠的数据注册表 - 即使在一个或几个节点关闭时也可以获得数据。

分布式应用程序提供了很多好处，但它们也抛出了一些复杂和难以解决的挑战。ZooKeeper框架提供了一个完整的机制来克服所有的挑战。竞争条件和死锁使用故障安全同步方法进行处理。另一个主要缺点是数据的不一致性，ZooKeeper使用原子性解析。

## ZooKeeper的好处

以下是使用ZooKeeper的好处：

简单的分布式协调过程

同步 - 服务器进程之间的相互排斥和协作。此过程有助于Apache HBase进行配置管理。

有序的消息

序列化 - 根据特定规则对数据进行编码。确保应用程序运行一致。这种方法可以在MapReduce中用来协调队列以执行运行的线程。

可靠性

原子性 - 数据转移完全成功或完全失败，但没有事务是部分的。

# Kafka

kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic zhuti

kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic zhongshu3partition

kafka-topics.bat --list --zookeeper localhost:2181

kafka-console-producer --broker-list localhost:9092 --topic zhuti

HIEL HITLER!!

kafka-console-consumer --zookeeper localhost:2181 --topic zhuti --from-beginning

RabbitMQ

Exchange交换机类型

Direct 、Fanout 、Topic三种类型，RabbitMQ默认有一个exchange，叫default exchange，它用一个空字符串表示，它是direct exchange类型。

WmsSerialNumber