



## 第二版：Dubbo 19 道

### 目录

第二版：Dubbo 19 道	1
1.Dubbo 是什么？	1
2.Dubbo 的使用场景有哪些？	1
3.Dubbo 核心功能有哪些？	2
4.Dubbo 核心组件有哪些？	2
5.Dubbo 服务器注册与发现的流程？	3
6.Dubbo 支持哪些协议，它们的优缺点有哪些？	3
7.Dubbo 推荐什么协议？	4
8.Dubbo 有哪些注册中心？	4
9.Dubbo 的注册中心集群挂掉，发布者和订阅者之间还能通信么？	4
10.Dubbo 使用的是什么通信框架？	5
11.Dubbo 集群提供了哪些负载均衡策略？	5
12.Dubbo 的集群容错方案有哪些？	5
13.Dubbo 支持哪些序列化方式？	6
14.Dubbo 超时设置有哪些方式？	6
15.服务调用超时会怎么样？	6
16.Dubbo 在安全方面有哪些措施？	7
17.Dubbo 类似的分布式框架还有哪些？	7
18.Dubbo 和 Spring Cloud 有什么关系？	7
19.Dubbo 和 Spring Cloud 有什么哪些区别？	7

我们的网站：<https://tech.souyunku.com>

关注我们的公众号：**搜云库技术团队**，回复以下关键字

回复：**进群** 邀请您进「技术架构分享群」

回复：**内推** 即可进：北京，上海，广州，深圳，杭州，成都，武汉，南京，

✧ 微信搜一搜

🔍 搜云库技术团队



郑州, 西安, 长沙「程序员工作内推群」

回复 **【1024】** 送 4000G 最新架构师视频

回复 **【PPT】** 即可无套路获取, 以下最新整理调优 PPT!

## 46 页《JVM 深度调优, 演讲 PPT》



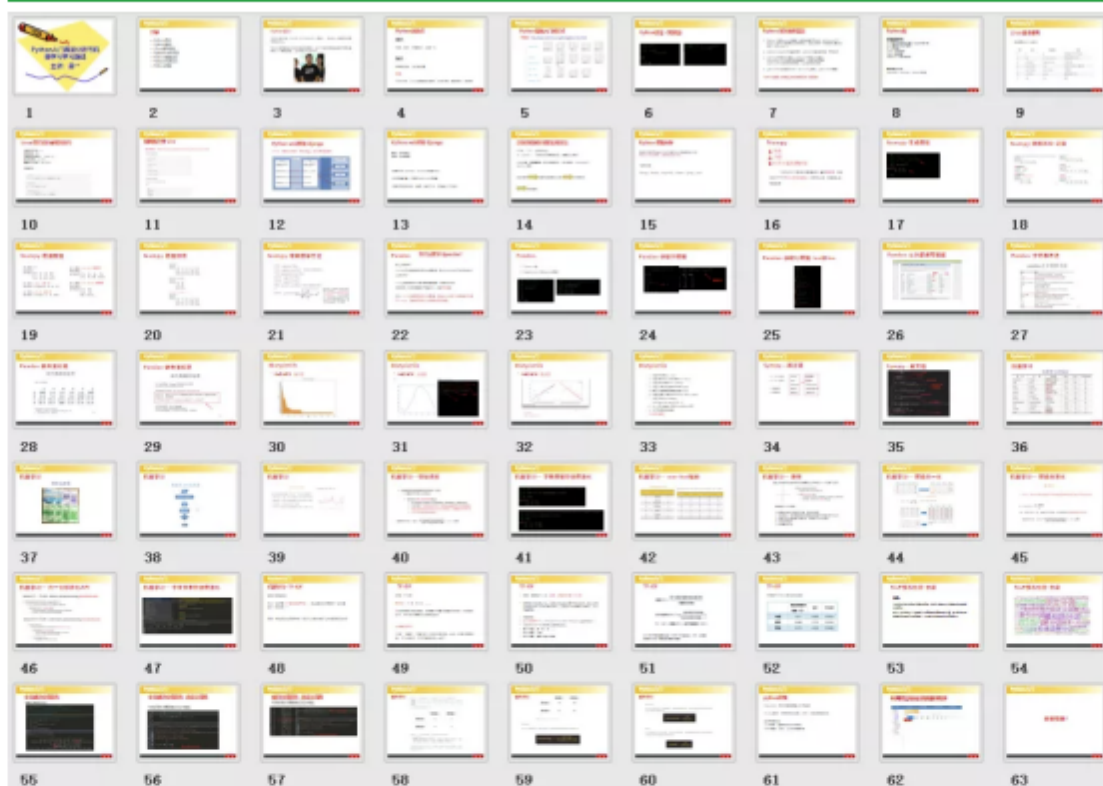
## 53 页《Elasticsearch 调优演讲 PPT》



## 63 页《Python 数据分析入门 PPT》

微信搜一搜

搜云库技术团队



微信扫一扫

<https://tech.souyunku.com>

技术、架构、资料、工作、内推  
专注于分享最有价值的互联网技术干货文章

## 1.Dubbo 是什么?



Dubbo 是一款高性能、轻量级的开源 RPC 框架，提供服务自动注册、自动发现等高效服务治理方案，可以和 Spring 框架无缝集成。

## 2.Dubbo 的使用场景有哪些？

- 透明化的远程方法调用：就像调用本地方法一样调用远程方法，只需简单配置，没有任何 API 侵入。
- 软负载均衡及容错机制：可在内网替代 F5 等硬件负载均衡器，降低成本，减少单点。
- 服务自动注册与发现：不再需要写死服务提供方地址，注册中心基于接口名查询服务提供者的 IP 地址，并且能够平滑添加或删除服务提供者。

## 3.Dubbo 核心功能有哪些？

- Remoting：网络通信框架，提供对多种 NIO 框架抽象封装，包括“同步转异步”和“请求-响应”模式的信息交换方式。
- Cluster：服务框架，提供基于接口方法的透明远程过程调用，包括多协议支持，以及软负载均衡，失败容错，地址路由，动态配置等集群支持。
- Registry：服务注册，基于注册中心目录服务，使服务消费方能动态的查找服务提供方，使地址透明，使服务提供方可以平滑增加或减少机器。

## 4.Dubbo 核心组件有哪些？

- Provider：暴露服务的服务提供方
- Consumer：调用远程服务消费方



- Registry: 服务注册与发现注册中心
- Monitor: 监控中心和访问调用统计
- Container: 服务运行容器

## 5.Dubbo 服务器注册与发现的流程?

- Provider (提供者) 绑定指定端口并启动服务。
- 提供者连接注册中心, 并发本机 IP、端口、应用信息和提供服务信息发送至注册中心存储。
- Consumer (消费者), 连接注册中心, 并发送应用信息、所求服务信息至注册中心。
- 注册中心根据消费者所求服务信息匹配对应的提供者列表发送至 Consumer 应用缓存。
- Consumer 在发起远程调用时基于缓存的消费者列表择其一发起调用。
- Provider 状态变更会实时通知注册中心、在由注册中心实时推送至 Consumer。

## 6.Dubbo 支持哪些协议, 它们的优缺点有哪些?

- Dubbo: 单一长连接和 NIO 异步通讯, 适合大并发小数据量的服务调用, 以及消费者远大于提供者。传输协议 TCP, 异步 Hessian 序列化。
- RMI: 采用 JDK 标准的 RMI 协议实现, 传输参数和返回参数对象需要实现 Serializable 接口, 使用 Java 标准序列化机制, 使用阻塞式短连接, 传输数据包大小混合, 消费者和提供者个数差不多, 可传文件, 传输协议 TCP。多个短连接 TCP 协议传输, 同步传输, 适用常规的远程服务调用和 RMI 互操作。在依赖低版本的 Common-Collections 包, Java 序列化存在安全漏洞。





- WebService: 基于 WebService 的远程调用协议, 集成 CXF 实现, 提供和原生 WebService 的互操作。多个短连接, 基于 HTTP 传输, 同步传输, 适用系统集成和跨语言调用。
- HTTP: 基于 Http 表单提交的远程调用协议, 使用 Spring 的 HttpInvoke 实现。多个短连接, 传输协议 HTTP, 传入参数大小混合, 提供者个数多于消费者, 需要给应用程序和浏览器 JS 调用。
- Hessian: 集成 Hessian 服务, 基于 HTTP 通讯, 采用 Servlet 暴露服务, Dubbo 内嵌 Jetty 作为服务器时默认实现, 提供与 Hessian 服务互操作。多个短连接, 同步 HTTP 传输, Hessian 序列化, 传入参数较大, 提供者大于消费者, 提供者压力较大, 可传文件。
- Memcache: 基于 Memcache 实现的 RPC 协议。
- Redis: 基于 Redis 实现的 RPC 协议。

## 7.Dubbo 推荐什么协议?

推荐使用 Dubbo 协议。

## 8.Dubbo 有哪些注册中心?

- Multicast 注册中心: Multicast 注册中心不需要任何中心节点, 只要广播地址, 就能进行服务注册和发现, 基于网络中组播传输实现。
- Zookeeper 注册中心: 基于分布式协调系统 Zookeeper 实现, 采用 Zookeeper 的 watch 机制实现数据变更。
- Redis 注册中心: 基于 Redis 实现, 采用 key/map 存储, key 存储服务名和类型, map 中 key 存储服务 url, value 服务过期时间。基于 Redis 的发布/订阅模式通知数据变更。
- Simple 注册中心。



## 9.Dubbo 的注册中心集群挂掉，发布者和订阅者之间还能通信么？

可以通讯。启动 Dubbo 时，消费者会从 Zookeeper 拉取注册的生产者的地址接口等数据，缓存在本地。每次调用时，按照本地存储的地址进行调用。

## 10.Dubbo 使用的是什么通信框架？

默认使用 Netty 作为通讯框架。

## 11.Dubbo 集群提供了哪些负载均衡策略？

- Random LoadBalance: 随机选取提供者策略，有利于动态调整提供者权重。截面碰撞率高，调用次数越多，分布越均匀。
- RoundRobin LoadBalance: 轮循选取提供者策略，平均分布，但是存在请求累积的问题。
- LeastActive LoadBalance: 最少活跃调用策略，解决慢提供者接收更少的请求。
- ConstantHash LoadBalance: 一致性 Hash 策略，使相同参数请求总是发到同一提供者，一台机器宕机，可以基于虚拟节点，分摊至其他提供者，避免引起提供者的剧烈变动。

默认为 Random 随机调用。

## 12.Dubbo 的集群容错方案有哪些？



- Failover Cluster: 失败自动切换, 当出现失败, 重试其它服务器。通常用于读操作, 但重试会带来更长延迟。
- Failfast Cluster: 快速失败, 只发起一次调用, 失败立即报错。通常用于非幂等性的写操作, 比如新增记录。
- Failsafe Cluster: 失败安全, 出现异常时, 直接忽略。通常用于写入审计日志等操作。
- Failback Cluster: 失败自动恢复, 后台记录失败请求, 定时重发。通常用于消息通知操作。
- Forking Cluster: 并行调用多个服务器, 只要一个成功即返回。通常用于实时性要求较高的读操作, 但需要浪费更多服务资源。可通过 `forks="2"` 来设置最大并行数。
- Broadcast Cluster: 广播调用所有提供者, 逐个调用, 任意一台报错则报错。通常用于通知所有提供者更新缓存或日志等本地资源信息。

默认的容错方案是 Failover Cluster。

### 13.Dubbo 支持哪些序列化方式?

默认使用 Hessian 序列化, 还有 Duddo、FastJson、Java 自带序列化。

### 14.Dubbo 超时设置有哪些方式?

Dubbo 超时设置有两种方式:

- 服务提供者端设置超时时间, 在 Dubbo 的用户文档中, 推荐如果能在服务端多配置就尽量多配置, 因为服务提供者比消费者更清楚自己提供的服务特性。





- 服务消费者端设置超时时间,如果在消费者端设置了超时时间,以消费者端为主,即优先级更高。因为服务调用方设置超时时间控制性更灵活。如果消费方超时,服务端线程不会定制,会产生警告。

## 15.服务调用超时会怎么样?

dubbo 在调用服务不成功时,默认是会重试两次。

## 16.Dubbo 在安全方面有哪些措施?

- Dubbo 通过 Token 令牌防止用户绕过注册中心直连,然后在注册中心上管理授权。
- Dubbo 还提供服务黑白名单,来控制服务所允许的调用方。

## 17.Dubbo 类似的分布式框架还有哪些?

比较著名的就是 Spring Cloud。

## 18.Dubbo 和 Spring Cloud 有什么关系?

Dubbo 是 SOA 时代的产物,它的关注点主要在于服务的调用,流量分发、流量监控和熔断。而 Spring Cloud 诞生于微服务架构时代,考虑的是微服务治理的方方面面,另外由于依托了 Spring、Spring Boot 的优势之上,两个框架在开始目标就不一致,Dubbo 定位服务治理、Spring Cloud 是打造一个生态。



## 19.Dubbo 和 Spring Cloud 有什么哪些区别？

Dubbo 底层是使用 Netty 这样的 NIO 框架，是基于 TCP 协议传输的，配合以 Hession 序列化完成 RPC 通信。

Spring Cloud 是基于 Http 协议 Rest 接口调用远程过程的通信，相对来说 Http 请求会有更大的报文，占的带宽也会更多。但是 REST 相比 RPC 更为灵活，服务提供方和调用方的依赖只依靠一纸契约，不存在代码级别的强依赖，这在强调快速演化的微服务环境下，显得更为合适，至于注重通信速度还是方便灵活性，具体情况具体考虑。