



哈尔滨工业大学 计算机科学与技术/国家示范性软件学院

# 携程Dubbo之路

杨大易

2021/12/14



# 为什么引入Dubbo?



- ❖ 从 2013 年底起，携程内主要使用的就是基于 HTTP 协议的 SOA 微服务框架。框架是携程内部自行研发的，受到当初设计的限制，框架本身的扩展性不是很好，使得用户要想自己扩展一些功能就会比较困难。
- ❖ 而Dubbo 作为一个高性能的 RPC 框架，不仅是一款业界知名的开源产品，它整体优秀的架构设计和数据传输方式可以解决上述这些问题。

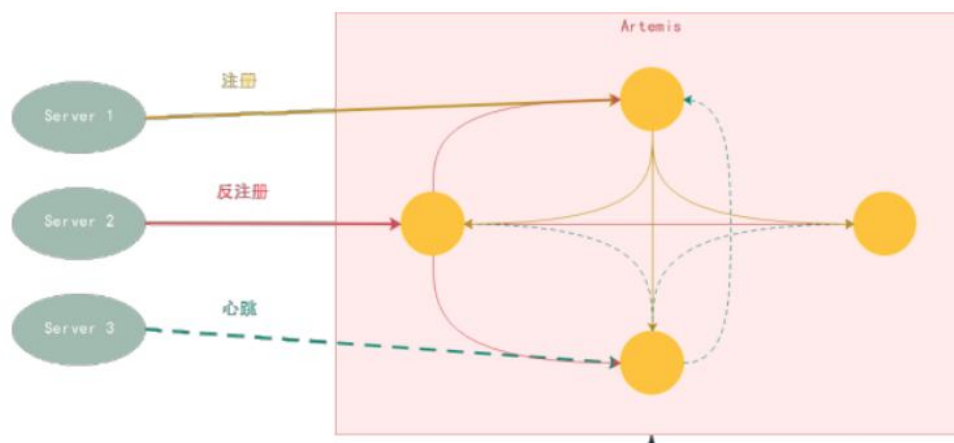


# Dubbo落地



## ❖ 服务治理

- 在服务治理这方面，携程现有的 SOA 框架已经有了一套完整的服务注册中心和服务治理系统。对于服务注册中心，大家比较常用的可能是 Apache Zookeeper。而我们使用的是参考 Netflix 开源的 Eureka 自行研发的注册中心 Artemis。



# Dubbo落地



## ❖ 服务监控

- 统计数据：对各种服务调用数据的定期汇总，比如调用量、响应时间、请求体和响应体的大小以及请求出现异常的情况等等。
- 监控服务：使用 **CAT**。**CAT** 是美团点评开源的一个实时的应用监控平台，它通过树形的 **Transaction** 和 **Event** 节点，可以将整个请求的处理过程记录下来。



# Dubbo落地



## ❖ 初版发布

- 在解决了服务治理和监控对接这两个问题后，完成了 **Dubbo** 在携程初步的一个本地化。在 2018 年 3 月，发布了 **Dubbo** 携程定制版的首个可用版本。
- 在正式发布前需要给这个产品起个新名字。既然是携程（Ctrip）加 **Dubbo**，就把这个定制版称为 **CDubbo**。

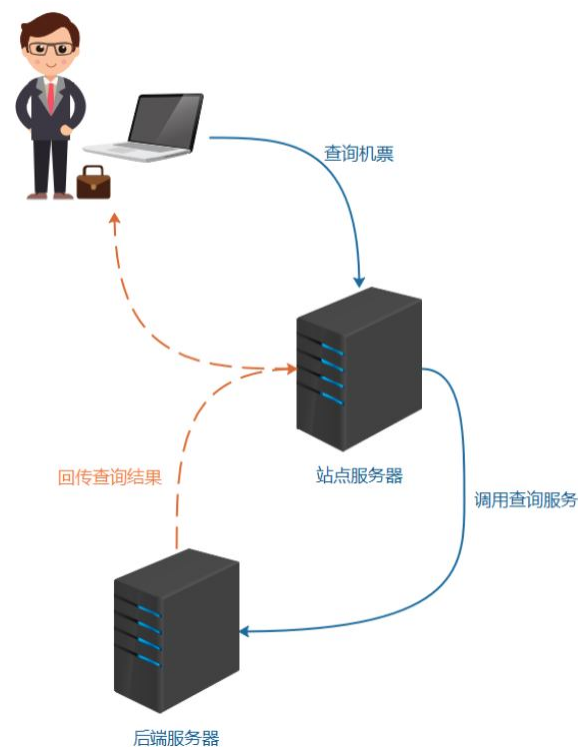
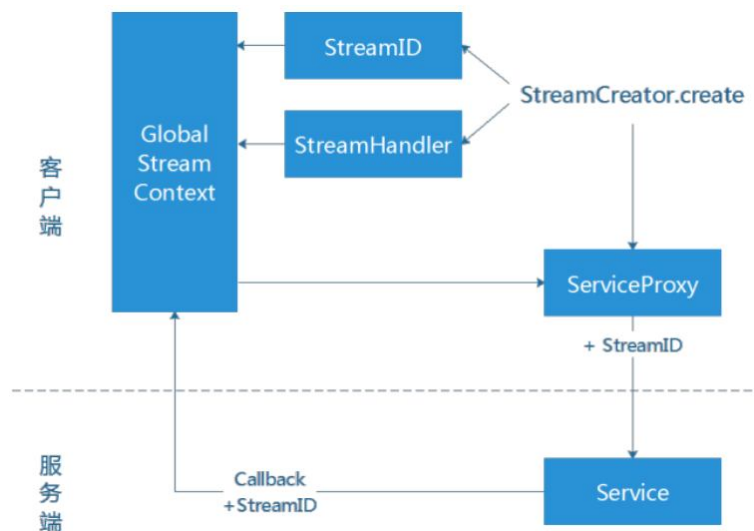


# Cdubbo功能扩展



## ❖ Callback 增强

- 对于全局唯一的callback接口实例，想要拿到请求上下文信息比较困难。
- 增加了 Stream 功能。



# Cdubbo功能扩展



## ❖ 序列化扩展

- 一些业务部门在之前开发 SOA 服务的时候，使用的是 **Google Protocol Buffer** 契约编写的请求数据模型。
- 为了便于将 SOA 服务迁移到 Dubbo，也在 Dubbo 中增加了 **GooglePB** 序列化方式的支持。
- 后续为了便于用户自行扩展，在 **PB** 序列化器的实现上增加了扩展接口，允许用户在外围继续增加数据压缩的功能。



# Cdubbo功能扩展



## ❖ 请求熔断

- 当客户端或服务端出现大范围请求出错或超时的时候，系统会自动执行 **fail-fast** 逻辑，不再继续发送和接受请求，而是直接返回错误信息。
- 使用业界成熟的解决方案：**Netflix** 开源的 **Hystrix** 。它不仅包含熔断的功能，还支持并发量控制、不同的调用间隔隔离等功能。
- **CDubbo**的服务端和客户端通过集成 **Hystrix** 来做请求的异常情况进行处理，避免发生雪崩效应。



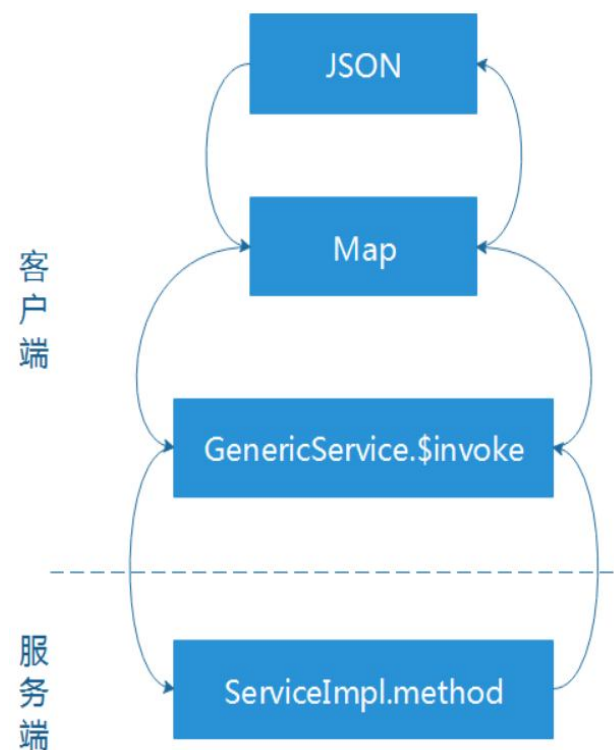


# Cdubbo功能扩展



## ❖ 服务测试工具

- 让测试人员在无需编写代码的前提下测试一个 **Dubbo** 服务，需要解决的三个问题：如何编写测试请求、如何发送测试请求和如何查看响应数据。
- 开发服务测试平台，用户可以在上面直接选择服务和实例，编写和发送测试请求。



# 后续功能规划



- ❖ **CDubbo** 一共发布了**27**个版本。携程的很多业务部门都已经接入了 **Dubbo** 。
- ❖ 在未来， **CDubbo** 还会扩展更多的功能，比如请求限流和认证授权等。





哈尔滨工业大学 计算机科学与技术/国家示范性软件学院

谢谢！