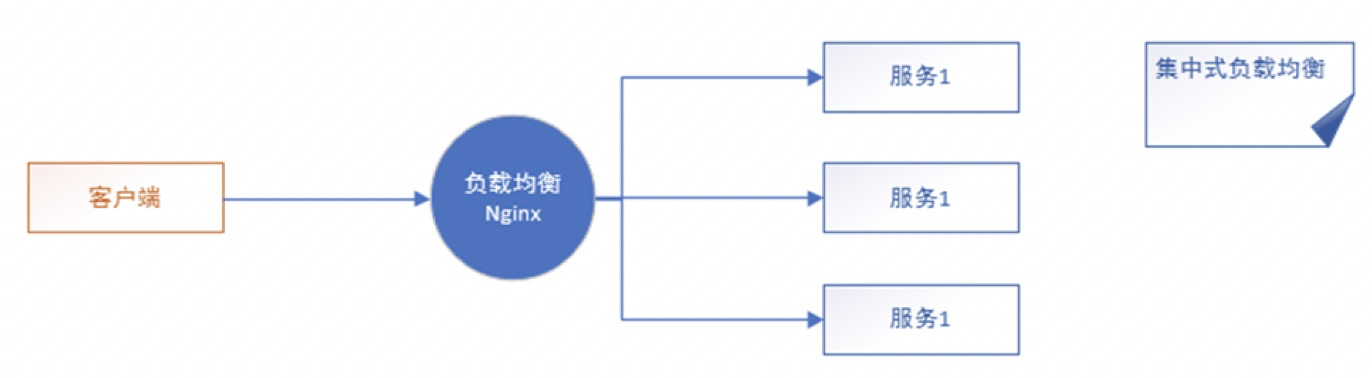
**[SpringCloud之Ribbon负载均衡配置](https://www.cnblogs.com/jing99/p/11612469.html)**

**一、负载均衡解决方案分类及特征**

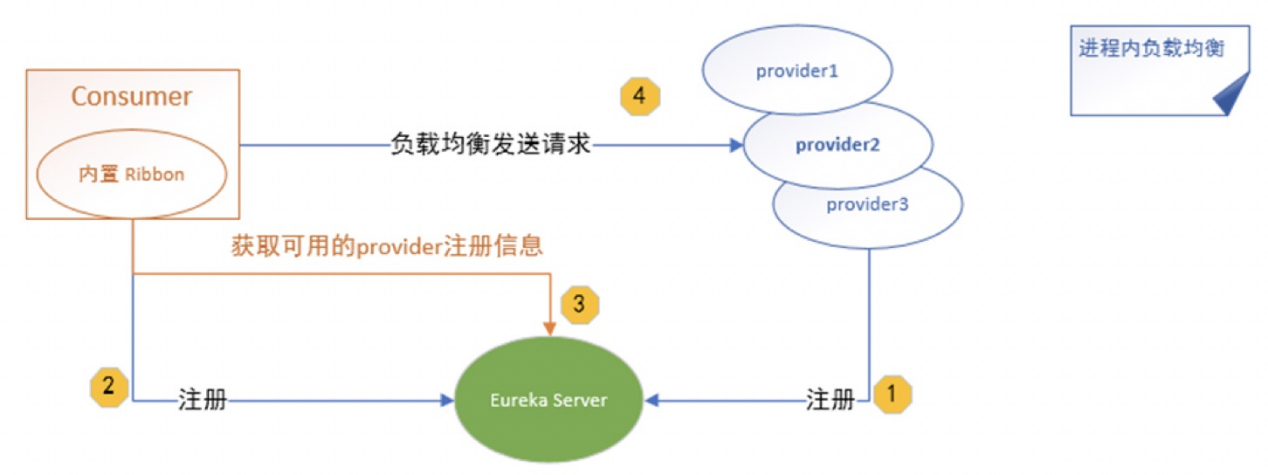
　　业界主流的负载均衡解决方案有：

**1.1 集中式负载均衡**

　　即在客户端和服务端之间使用独立的负载均衡设施(可以是硬件，如F5, 也可以是软件，如nginx), 由该设施负责把访问请求通过某种策略转发至服务端。  
　　　　　　　　　　　　

**1.2 进程内负载均衡**

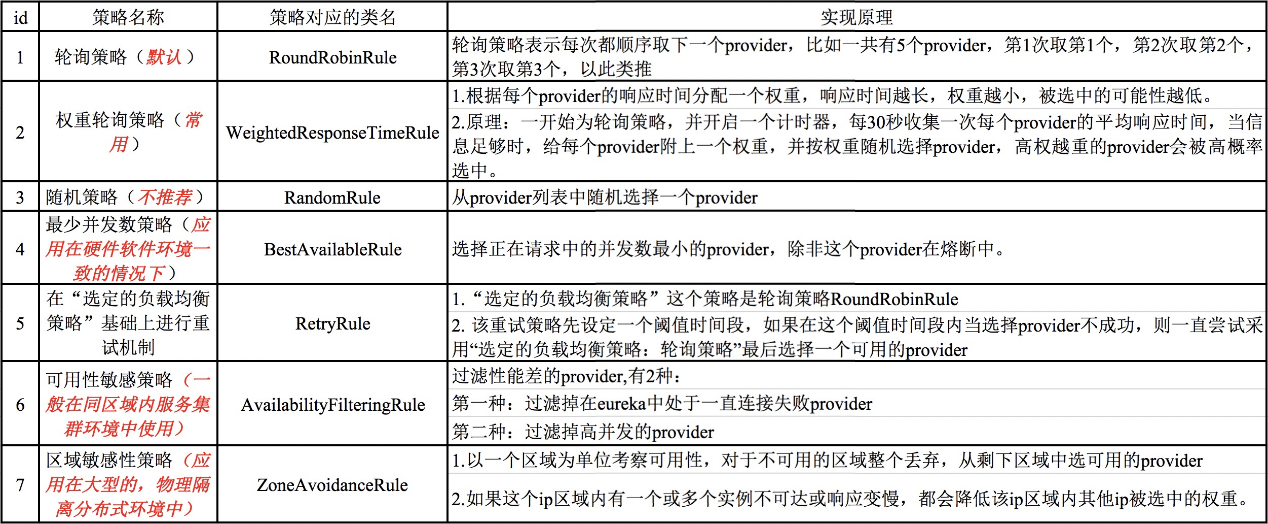
　　将负载均衡逻辑集成到客户端组件中，客户端组件从服务注册中心获知有哪些地址可用，然后自己再从这些地址中选择出一个合适的服务端发起请求。Ribbon就是一个进程内的负载均衡实现。



**二、Ribbon常用负载均衡策略**

　　Ribbon就属于进程内负载均衡，它只是一个类库，**集成于Eureka Client进程**，Eureka Client进程通过访问注册中心Eureka Server发现服务列表，发现的服务列表信息是由ribbon来管理的。当访问Application Service的时候，Application Client会通过ribbon来找到合适的Application Service地址信息，并发起远程调用请求。

　　Ribbon的负载均衡策略是通过不同的类型来实现的，下表详细介绍一些常用负载均衡策略及对应的Ribbon策略类。



**三、指定负载均衡策略**

　　可以通过全局配置文件来改变当前环境中使用的Ribbon负载均衡策略。在Ribbon中没有通用的负载均衡策略配置方案。

#设置负载均衡策略 eureka-application-service为调用的服务的名称

eureka-application-service.ribbon.NFLoadBalancerRuleClassName=com.netflix.loadbalancer.RandomRule

**四、 点对点直连测试**

　　在商业开发中，经常会更新或升级部分服务。当系统某服务出现bug后，需要修改并再次测试。如果还是通过生产环境中的注册中心来注册发现服务，那么影响面太大。这个时候可以考虑将Application Service注册到一个测试Eureka Server上，使用点对点直连的方式让Application Client直接访问Application Service，打到测试的目的。  
　　使用点对点直连测试很容器。只要在启动的时候避免注册和发现服务（删除启动类上的@EnableEurekaClient注解），并在全局配置文件中配置下述内容即可。（全局配置文件中关于Eureka Server的配置可删除）。

[复制代码](javascript:void(0);)

spring.application.name=eureka-application-client

server.port=8080

# 点对点直连是不发现服务，不是不注册服务。

# 任何Eureka Client都必须注册。如果没有配置Eureka Server节点列表，则注册失败。Eureka client无法正常启动。

eureka.client.serviceUrl.defaultZone=http://eurekaserver1:111111@eurekaserver1:8761/eureka/,http://eurekaserver2:222222@eurekaserver2:8761/eureka/

#点对点直连测试配置

# 关闭ribbon访问注册中心Eureka Server发现服务，但是服务依旧会注册。

ribbon.eureka.enabled=false

# 配置服务列表，其中eureka-application-service代表要访问的服务的应用名，如果有多个服务结点组成集群，多个节点的配置信息使用逗号','分隔。

# 配置服务列表，需要配置要调用的服务的名字和服务所在的位置。

# 服务的名字，就是Application Service中配置的spring.application.name。

# 服务的位置，就是服务的所在ip和端口。

# 如果服务位置有多个，也就是服务集群，那么使用逗号','分割多个服务列表信息。

eureka-application-service.ribbon.listOfServers=localhost:8083