

服务生产与消费

Eureka 服务治理体系中有 3 个核心角色：服务注册中心、服务提供者、服务消费者。服务注册中心及服务提供者，前篇已做介绍。

本章主要介绍 Eureka 服务治理体系中的服务消费，以 Spring Cloud 服务调用中的 Ribbon+RestTemplate 方式为主。

什么是 Ribbon

Ribbon 是 Netflix 发布的开源项目，主要功能是提供客户端的软件负载均衡算法，将 Netflix 的中间层服务连接在一起。Ribbon 客户端组件提供一系列完善的配置项如连接超时，重试等。简单的说，就是在配置文件中列出 Load Balancer（简称 LB）后面所有的机器，Ribbon 会自动的帮助你基于某种规则（如简单轮询，随即连接等）去连接这些机器。我们也很容易使用 Ribbon 实现自定义的负载均衡算法。

Ribbon 的负载均衡策略

1. 简单轮询负载均衡

以轮询的方式依次将请求调度不同的服务器，即每次调度执行 $i = (i + 1) \bmod n$ ，并选出第 i 台服务器。

2. 随机负载均衡

随机选择状态为 UP 的 Server。

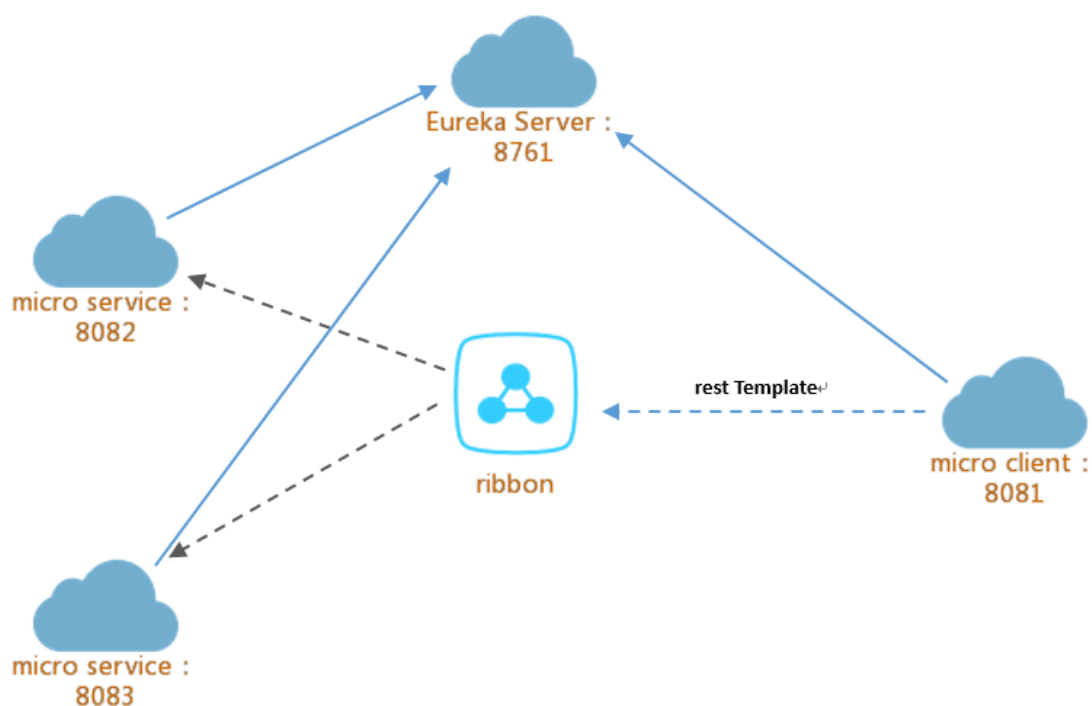
3. 加权响应时间负载均衡

根据相应时间分配一个 weight，相应时间越长，weight 越小，被选中的可能性越低。

4. 区域感知轮询负载均衡

复合判断 server 所在区域的性能和 server 的可用性选择 server。

实现原理



- 一个服务注册中心，eureka server，端口为 8761。
- micro-service 工程跑了两个副本，端口分别为 8082、8083，分别向服务注册中心注册。
- micro-client 端口为 8081，向服务注册中心注册。
- 当 micro-client 通过 restTemplate 调用 micro-service 的 hello 接口时，因为用 ribbon 进行了负载均衡，会轮流的调用 micro-service : 8082 和 8083 两个端口的 hello 接口。