为什么Ribbon在新版本的Spring框架中被移除了呢？一起探讨一下

写在最前面

众所周知，技术更新迭代速度将会越来越快，AI加持以后的技术环境将会更加变化多端，也许我们可以守旧，但是我们不能不跟上技术发展步伐，发展意味着进步，进步意味着总有新技术进来或发明，意味着旧的技术必将退出历史舞台。以下针对我们最近在进行框架升级时碰到的问题，引发的一些研究和思考：

Ribbon移除的主要原因，主要如下：

1. Ribbon已经停止维护：Ribbon作为Netflix开源的一个组件，早已进入维护状态。这意味着它不再接受新的功能开发，且对于已知的问题可能无法得到及时的修复。因此，继续使用Ribbon可能会带来潜在的风险和不确定性。
2. Spring Cloud的整合需求：Spring Cloud作为一个微服务框架，需要与其生态系统中的其他组件进行紧密的整合。而Ribbon可能与Spring Cloud的一些新特性和发展方向不完全契合，因此为了更好地推进Spring Cloud的发展，选择移除Ribbon并寻找一个更合适的负载均衡解决方案是必要的。

Spring Cloud LoadBalancer作为接班人

它的核心优势及技术特点如下：

1. 支持响应式编程：采用Spring WebFlux框架实现客户端负载均衡调用，可以更好地应对高并发场景。
2. 统一的配置管理：作为Spring Cloud官方提供的客户端负载均衡器，可以与Spring Cloud其他组件无缝集成，实现统一的配置管理和监控。
3. 更好的可扩展性：提供了丰富的扩展点，开发者可以根据实际需求自定义负载均衡策略和过滤器，实现更加灵活和可扩展的负载均衡功能。
4. Spring Cloud LoadBalancer还支持常见的负载均衡算法，如随机选择和轮询选择等，并且可以通过实现自定义的LoadBalancer接口来扩展其功能。

从技术的角度，我们来深入浅出认识一下**Spring Cloud LoadBalancer**

Spring Cloud LoadBalancer是Spring Cloud中的一个重要组件，用于实现客户端的负载均衡。在微服务架构中，服务通常会被部署在多个实例上，以提高可用性和可扩展性。客户端在调用这些服务时，需要有一种机制来均匀地将请求分发到这些实例上，这就是负载均衡器的作用。

核心概念和组件：

---

1. **ReactiveLoadBalancer**：这是一个反应式的负载均衡器接口，它负责根据一定的策略从多个服务实例中选择一个合适的服务实例来处理请求。
2. **ServiceInstanceListSupplier**：这个接口负责提供可用的服务实例列表。通常，这些实例信息会从服务注册中心（如Eureka、Consul、Nacos等）获取。
3. **负载均衡策略**：Spring Cloud LoadBalancer提供了多种内置的负载均衡策略，如轮询（Round Robin）、随机（Random）等。此外，它还支持自定义负载均衡策略，以满足特定的业务需求。

工作原理：

---

1. 当客户端发起一个服务调用请求时，Spring Cloud LoadBalancer会被触发。
2. LoadBalancer首先通过ServiceInstanceListSupplier获取当前可用的服务实例列表。
3. 接下来，LoadBalancer会根据配置的负载均衡策略（如轮询、随机等）从服务实例列表中选择一个合适的服务实例。
4. 一旦选择了服务实例，LoadBalancer就会将请求重定向到该实例上。
5. 如果选定的服务实例不可用或响应时间过长，LoadBalancer还可以根据配置的重试机制尝试其他服务实例。

特性与优势：

---

1. **灵活性**：支持多种负载均衡策略，并可以根据业务需求自定义策略。
2. **可扩展性**：可以轻松地与现有的微服务架构集成，并支持动态扩展服务实例。
3. **高可用性**：通过智能地分发请求到不同的服务实例，确保系统的高可用性。
4. **易于监控和管理**：与Spring Cloud的其他组件无缝集成，便于统一监控和管理。

使用场景：

---

* 当服务部署在多个实例上时，确保请求能够均匀分发到这些实例上。
* 在服务调用过程中实现故障转移和容错处理。
* 根据业务需求实现特定的负载均衡策略，如加权轮询、基于响应时间的负载均衡等。

最后

综上所述，Ribbon被新版本Spring移除的原因主要是其已停止维护和与Spring Cloud发展方向的不完全契合。而替代的方案是Spring Cloud LoadBalancer，它具有响应式编程支持、统一配置管理和更好的可扩展性等优势。

关于以上技术点，你有什么想说的吗？评论区我们一起讨论一下，讨论和争议是进步最快的学习方式。

原文链接: https://juejin.cn/post/7357728382316871690