**Spring Cloud 组件**

**Spring Cloud Config:管理配置工具**，支持使用Git存储配置内容，可以使用它实现应用配置的外部化存储，并支持客户端配置信息刷新、加密/解密配置内容等

**Spring Cloud Netflix:核心组件**，对多个Netflix OSS开源套件进行整合

**Eureka:服务治理组件**，包含服务注册中心、服务注册与发现机制的实现

**Hystrix:容错管理组件**，实现断路器模式，帮助服务依赖中出现的延迟和故障提供强大的容错能力

**Ribbon:客户端负载均衡的服务调用组件**

**Feign:基于Ribbon和Hystrix的声明式服务调用组件**

**Zuul:网关组件**，提供智能路由、访问过滤等功能

**Archaius:**外部化配置文件

**Spring Cloud Bus:事件、消息总线**，用于传播集群中的状态变化或事件，以触发后续的处理，比如用来动态刷新配置等

**Spring Cloud Cluster**:针对ZooKeeper、Redis、Hazelcast、Consul的选举算法和通用状态模式的实现

**Spring Cloud Cloudfoundry**:与Pirvotal Coudfoundry的整合支持

**Spring Cloud Consul:服务发现与配置管理工具**

**Spring Cloud Stream:**通过Redis、Rabbit或者Kafka实现的消息微服务，可以通过简单的声明式模型来发生和接收消息

**Spring Cloud AWS:**用于简化整合Amazon Web Service的组件

**Spring Cloud Security:**安全工具包，提供在Zuul代理中对OAuth2客户端请求的中继器

**Spring Cloud Sleuth:** Spring Cloud应用的分布式跟踪实现，可以完美整合Zipkin

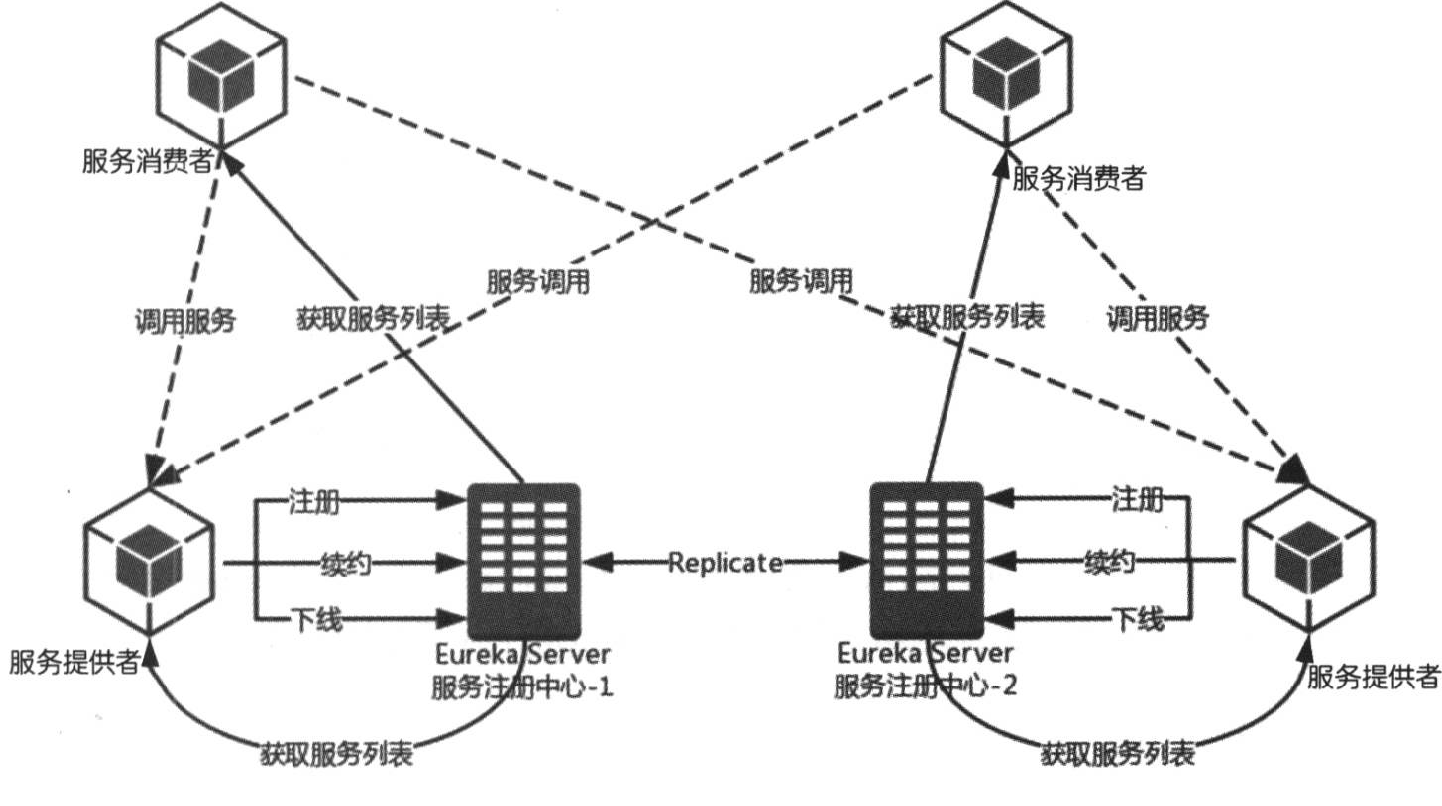
**Spring Cloud ZooKeeper:**基于ZooKeeper的服务发现与配置管理组件

**Spring Cloud Starters:** Spring Cloud的基础组件，它是基于Spring Boot风格项目的基础依赖模块

**Spring Cloud CLI：**用于在Groovy中快速创建Spring Cloud应用的Spring Boot CLI插件

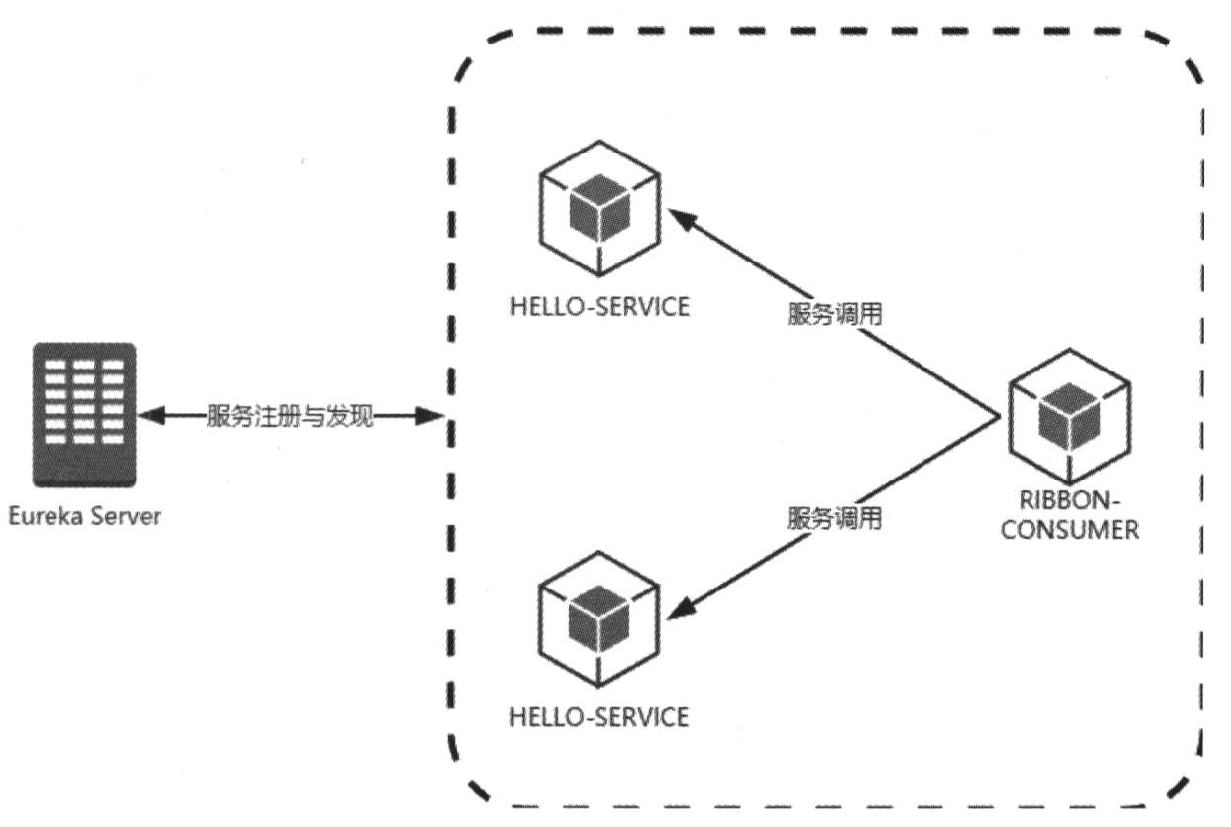
**Eureka:服务治理组件**

………………….



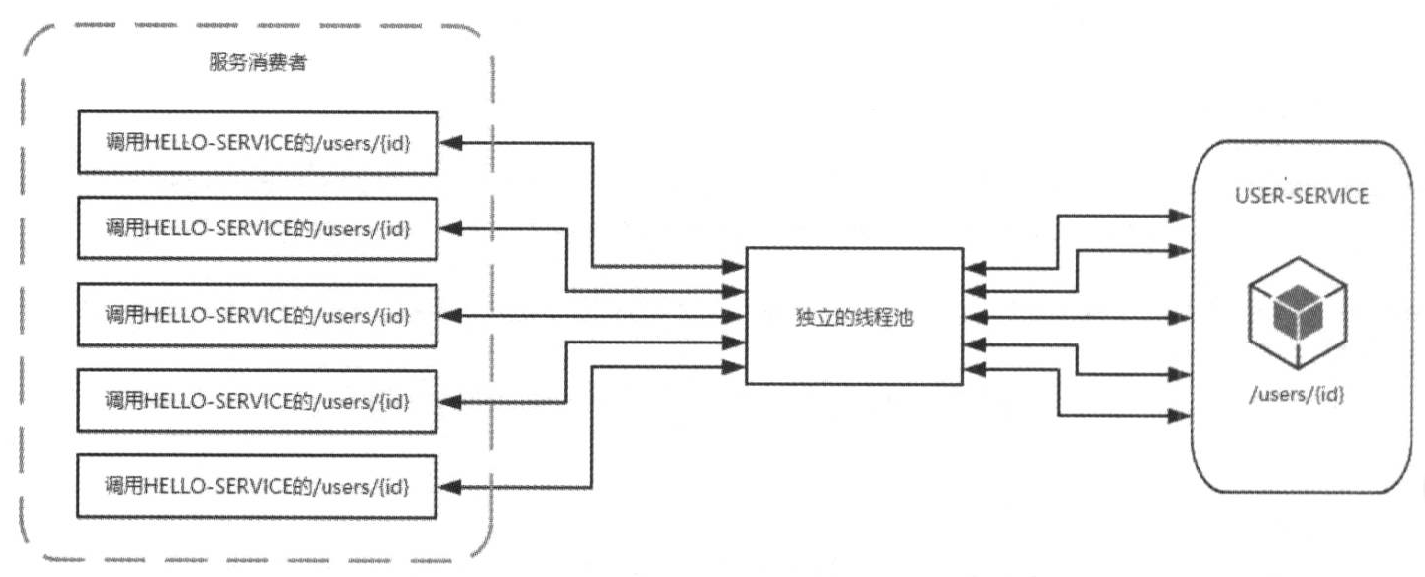
**服务容错保护: Spring Cloud Hystrix**

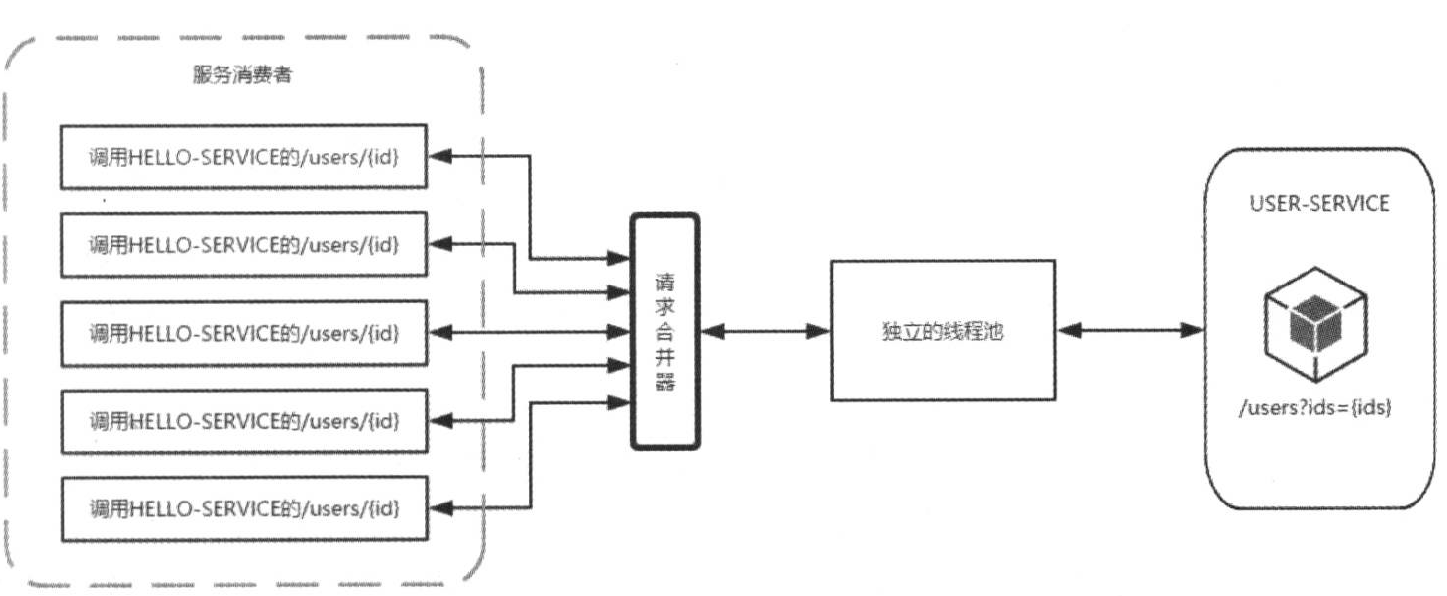
在微服务架构中，我们将系统拆分成了很多服务单元，各单元的应用间通过服务注册与订阅的方式互相依赖。由于美国系统都在不同的进程中运行，依赖通过远程调用的方式执行，这样就有可能因为网络原因或是依赖服务自身问题出现调用故障或延迟，而这些问题会直接导致调用方的对外服务也出现延迟，若此时调用方的请求不断增加，最好就好因等待出现故障的依赖方响应形成任务积压，最终导致自身服务的瘫痪



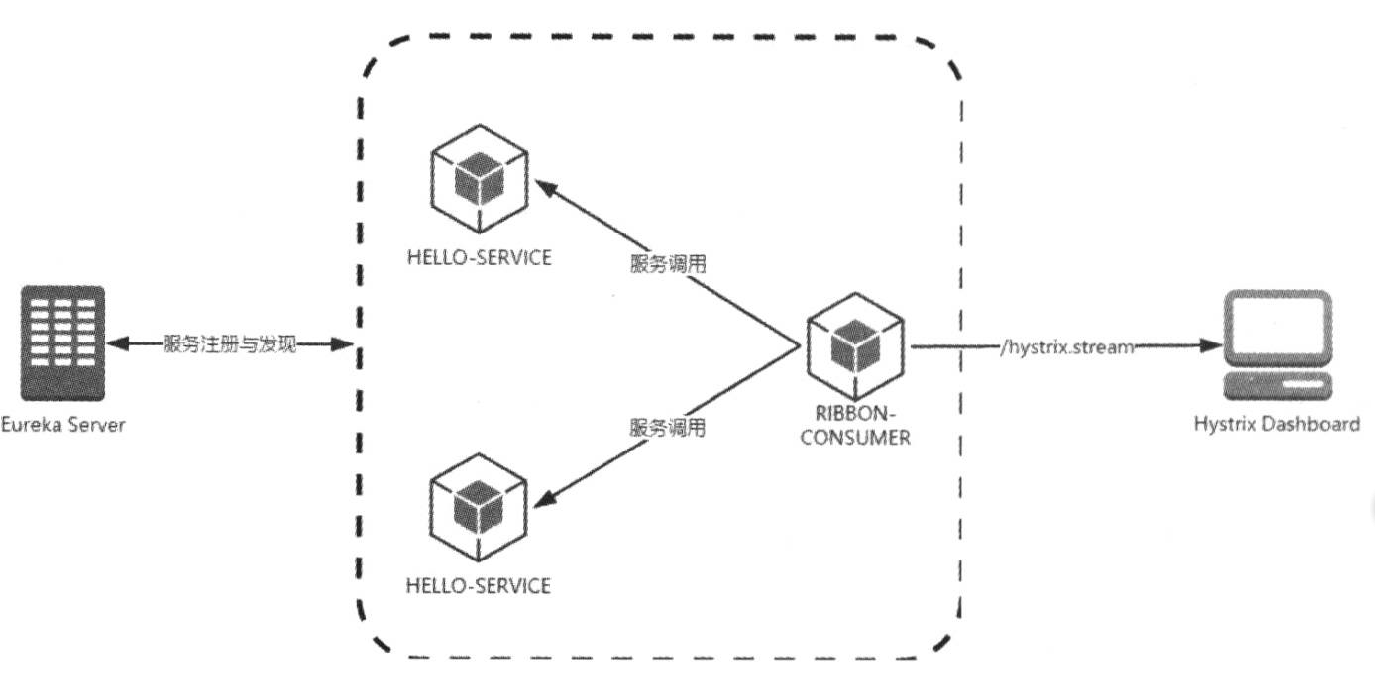
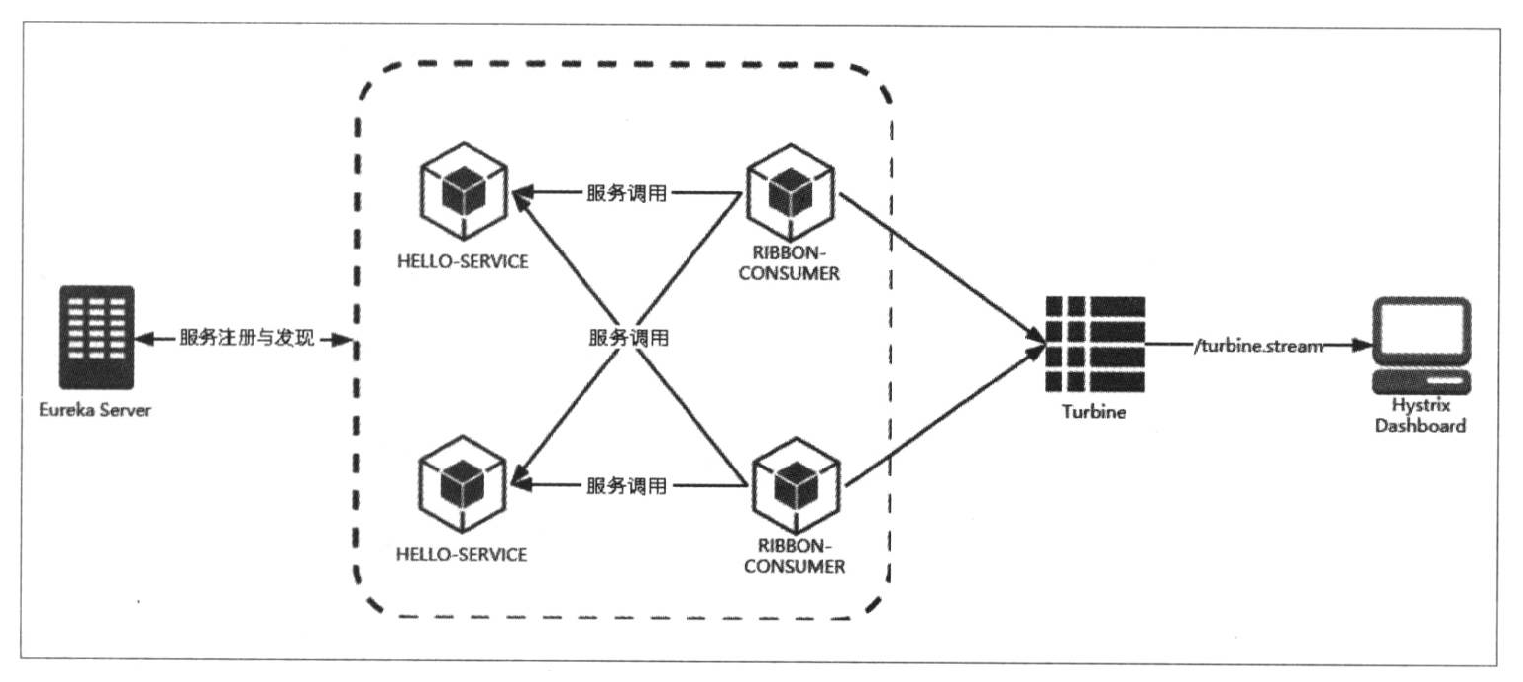
**请求合并:**

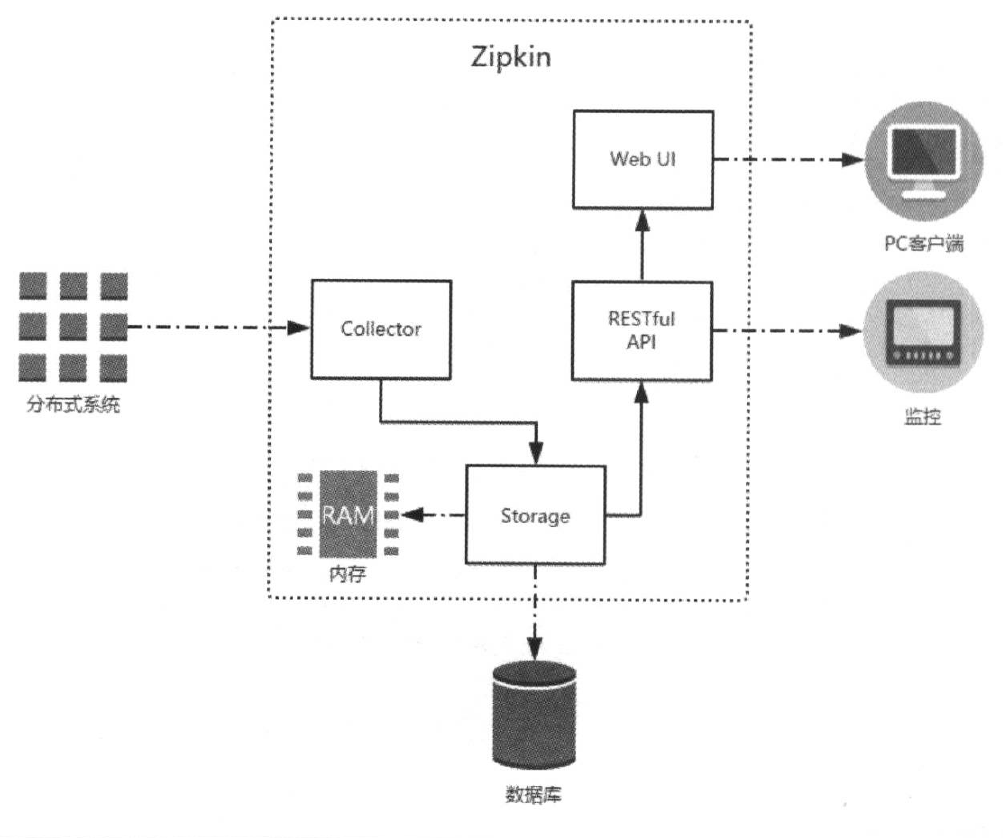
通过请求合并可以减少请求的数量以缓解依赖服务线程池的资源，但是在使用的时候也需要注意它所带来的额外开销:用于请求合并的延迟时间窗会使得依赖服务的请求延迟增高。





**监控:**



**API网关: Spring Cloud Zuul**

