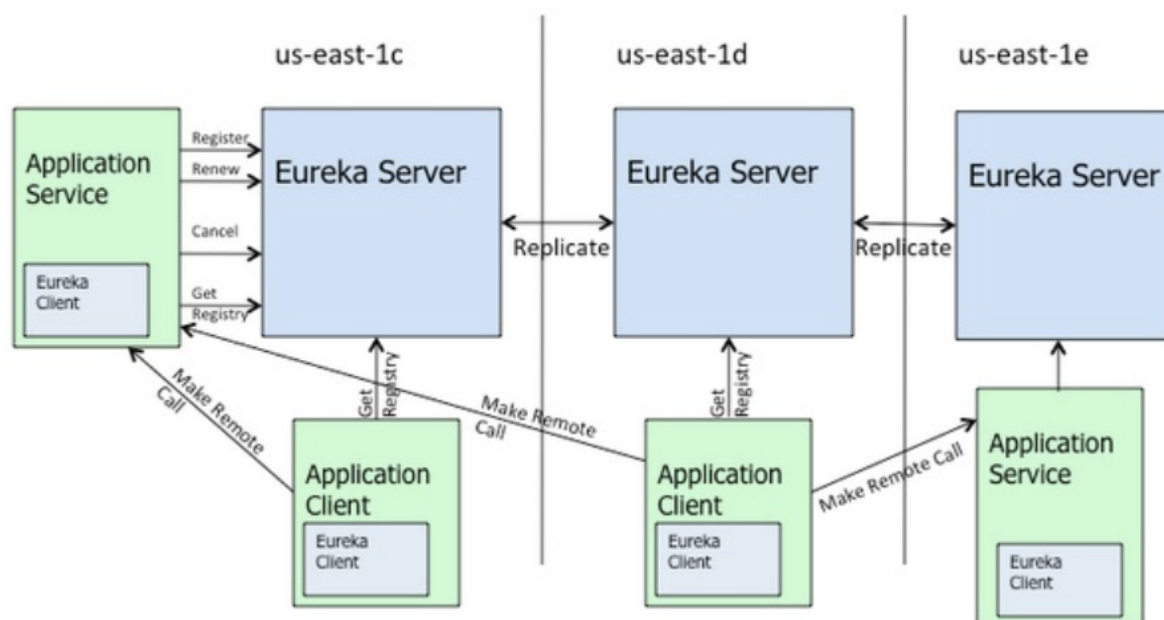


摘自: <https://www.cnblogs.com/ilinuxer/p/6580998.html>

SpringCloud分布式开发

- 1、服务发现 -- Netflix Eureka
- 2、客户端负载均衡 -- Netflix Ribbon
- 3、断路器 -- Netflix Hystrix
- 4、服务网关 -- Netflix Zuul
- 5、分布式配置 -- Spring Cloud Config

Eureka

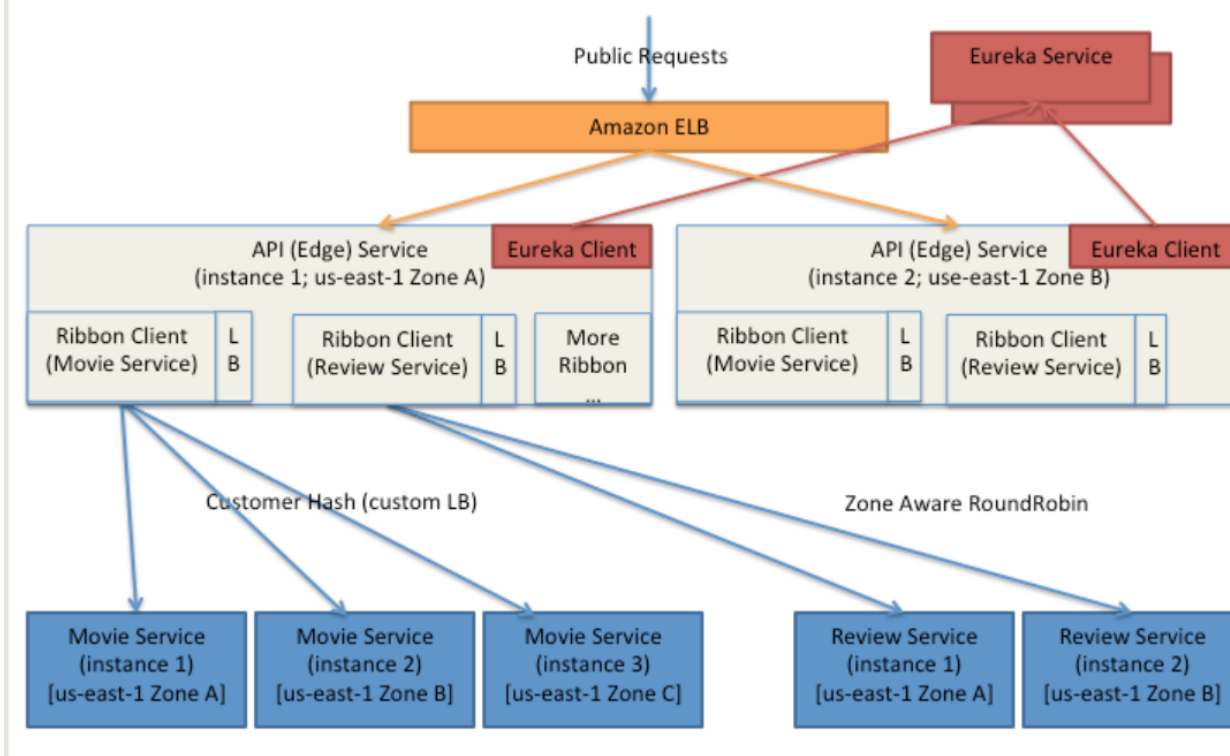


<http://www.ityouknow.com/springcloud/2017/05/12/eureka-provider-consumer.html>

一个RESTful服务，用来定位运行在AWS地区（Region）中的中间层服务。由两个组件组成：Eureka服务器和Eureka客户端。Eureka服务器用作注册服务器。Eureka客户端是一个java客户端，用来简化与服务器的交互、作为轮询负载均衡器，并提供服务的故障切换支持。Netflix在其生产环境中使用的是另外的客户端，它提供基于流量、资源利用率以及出错状态的加权负载均衡。

Ribbon

Ribbon，主要提供客户侧的软件负载均衡算法。



Ribbon客户端组件提供一系列完善的配置选项，比如连接超时、重试、重试算法等。

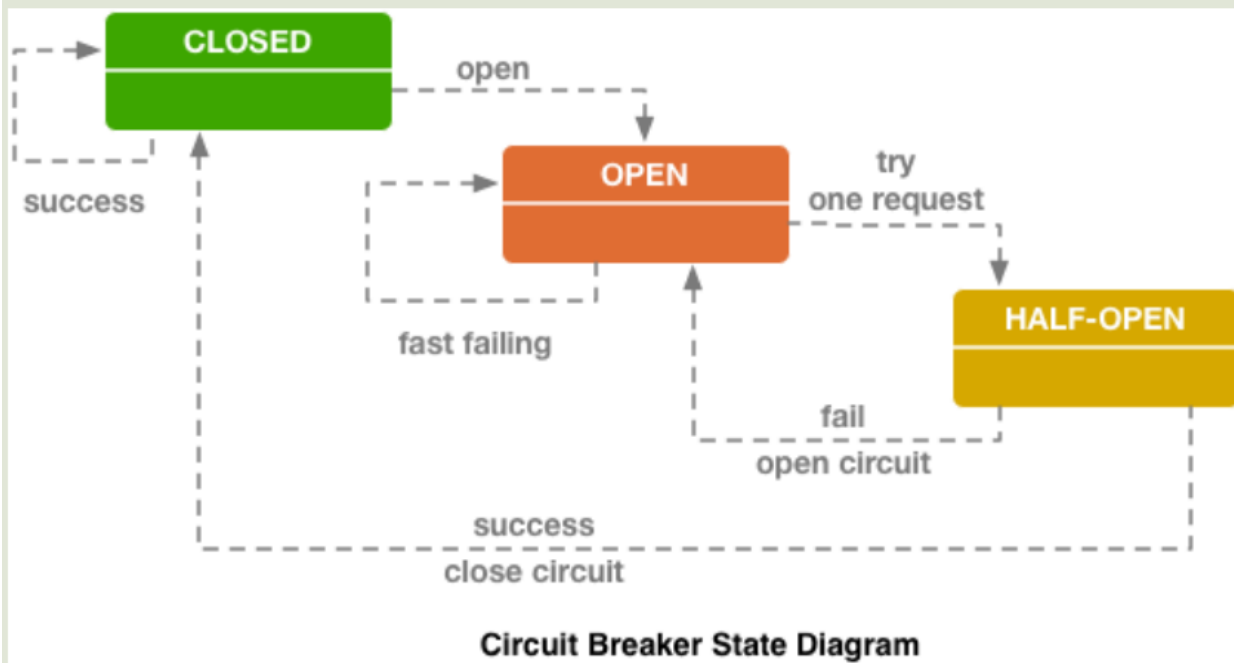
Ribbon内置可插拔、可定制的负载均衡组件。下面是用到的一些负载均衡策略：

- 简单轮询负载均衡
- 加权响应时间负载均衡
- 区域感知轮询负载均衡
- 随机负载均衡

Ribbon中还包括以下功能：

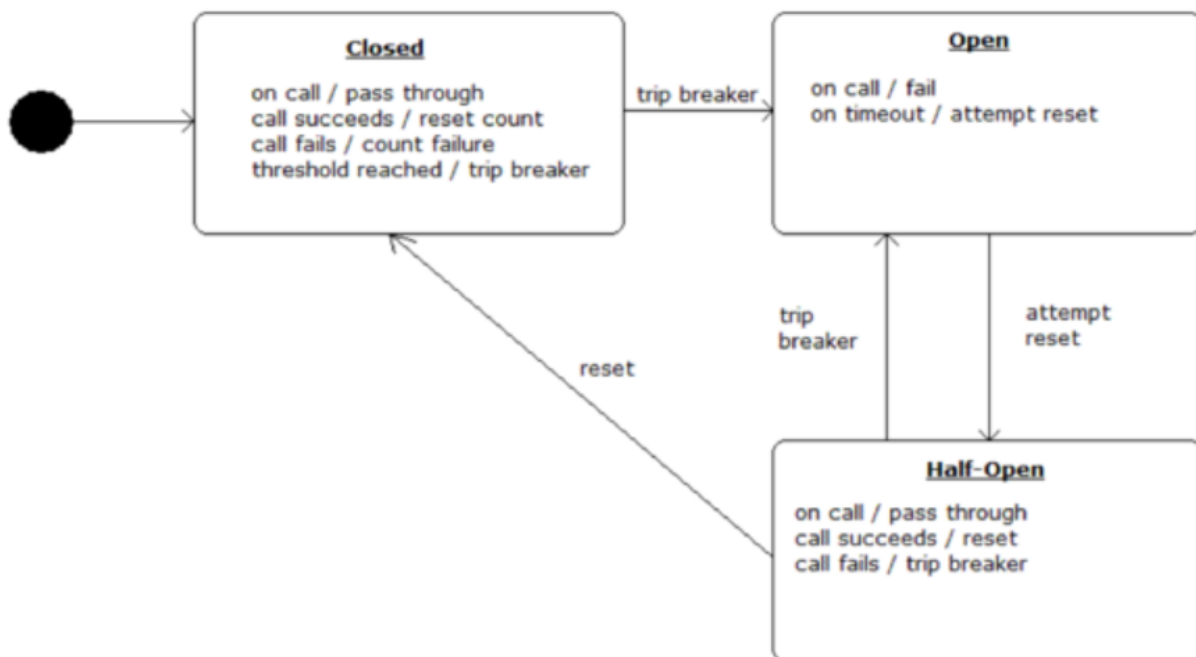
- 易于与服务发现组件（比如Netflix的Eureka）集成
- 使用Archaius完成运行时配置
- 使用JMX暴露运维指标，使用Servo发布
- 多种可插拔的序列化选择
- 异步和批处理操作（即将推出）
- 自动SLA框架（即将推出）
- 系统管理/指标控制台（即将推出）

Hystrix



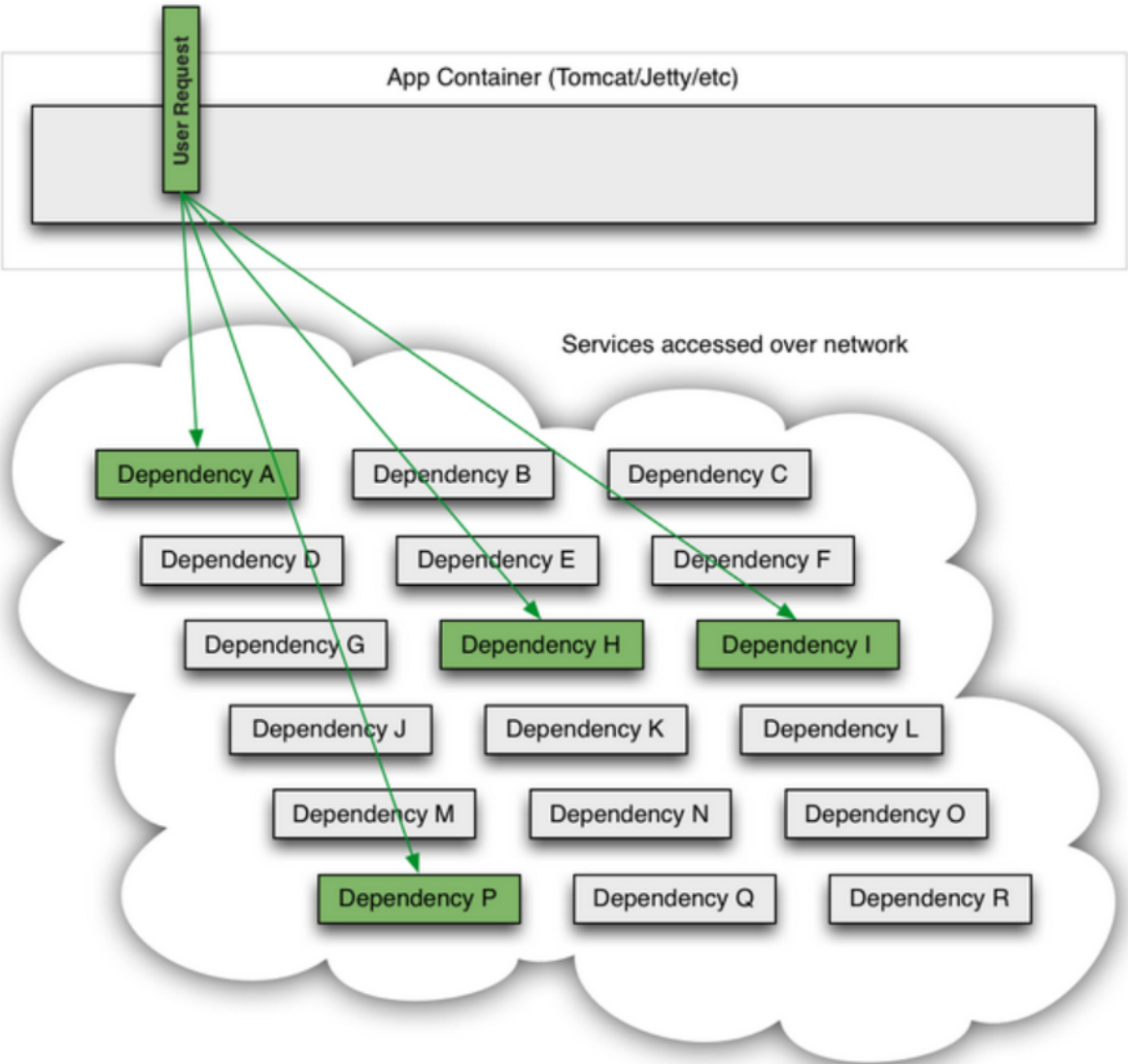
<http://www.ityouknow.com/springcloud/2017/05/16/springcloud-hystrix.html>

断路器可以防止一个应用程序多次试图执行一个操作，即很可能失败，允许它继续而不等待故障恢复或者浪费 CPU 周期，而它确定该故障是持久的。断路器模式也使应用程序能够检测故障是否已经解决。如果问题似乎已经得到纠正，应用程序可以尝试调用操作。

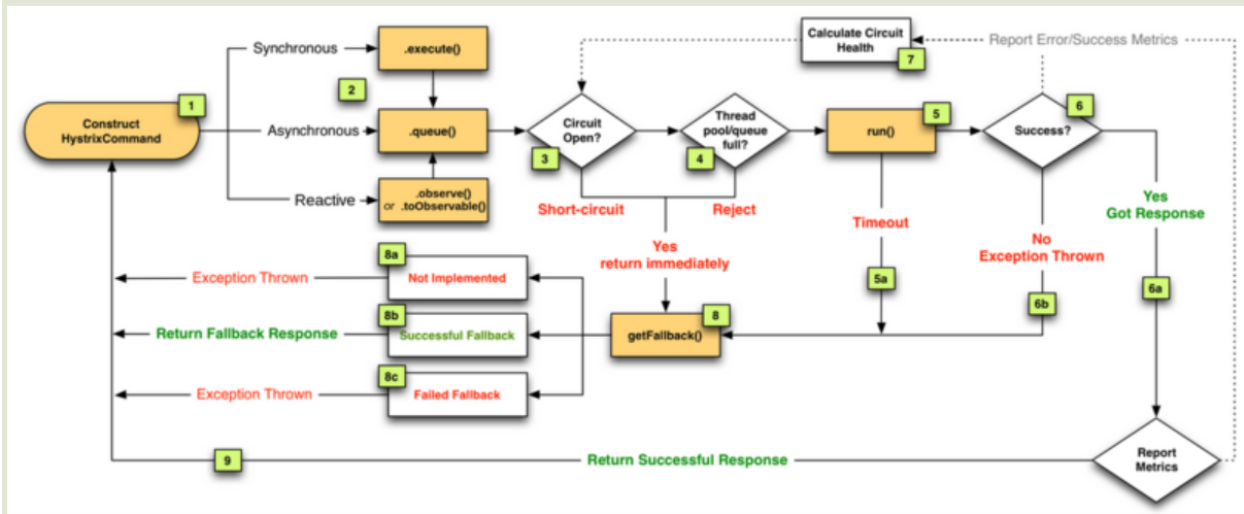


断路器增加了稳定性和灵活性，以一个系统，提供稳定性，而系统从故障中恢复，并尽量减少此故障的对性能的影响。它可以帮助快速地拒绝对一个操作，即很可能失败，而不是等待

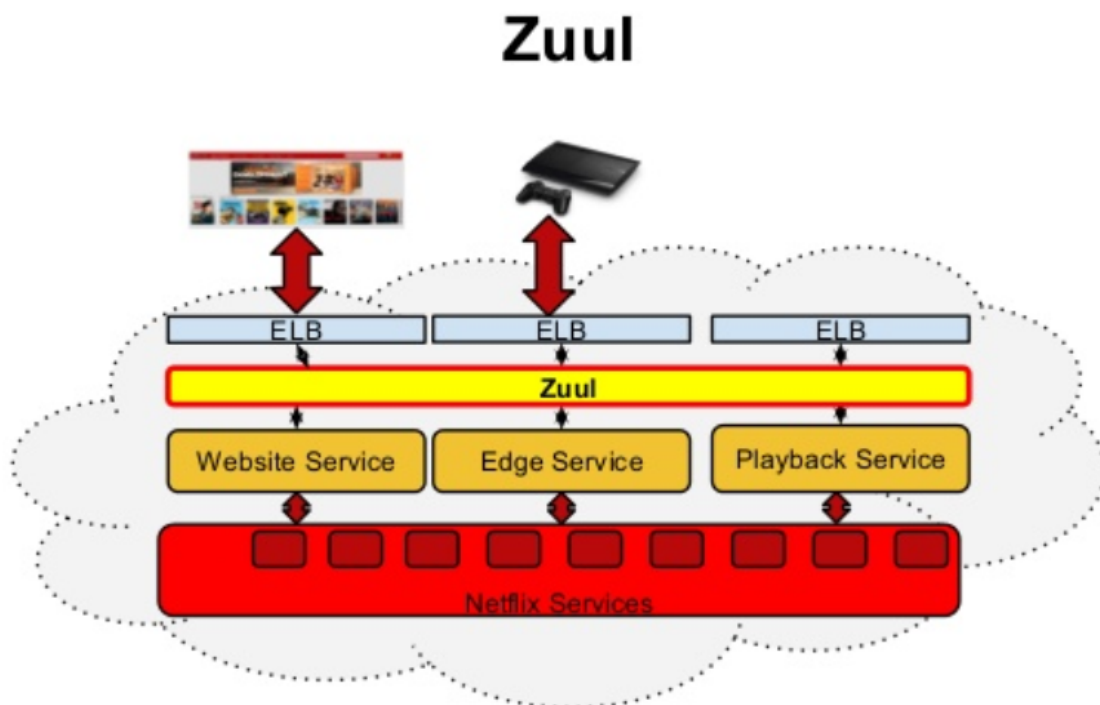
操作超时（或者不返回）的请求，以保持系统的响应时间。如果断路器提高每次改变状态的时间的事件，该信息可以被用来监测由断路器保护系统的部件的健康状况，或以提醒管理员当断路器跳匣，以打开状态



流程图



Zuul



类似Nginx，反向代理的功能，不过netflix自己增加了一些配合其他组件的特性。

Spring Cloud Config

ConfigServer.java:

```
@SpringBootApplication
@EnableConfigServer
public class ConfigServer {...}
```

Spring Cloud
Config Server



GIT



SVN



File

这个还是静态的，得配合Spring Cloud Bus实现动态的配置更新。