Dubbo +Nacos

服务治理重新实现

小马哥（ mercyblitz ）

自我介绍

小马哥（mercyblitz）

- Java 劝退师
- Apache Dubbo PPMC
- Spring Cloud Alibaba 架构师

为何劝退

- Oracle 说，Java 收费
- 技术太多，害怕掉队
- 时间太少，如何把妹

主要议程

- 服务治理基础组件
- Nacos 简介
- 服务治理重新实现

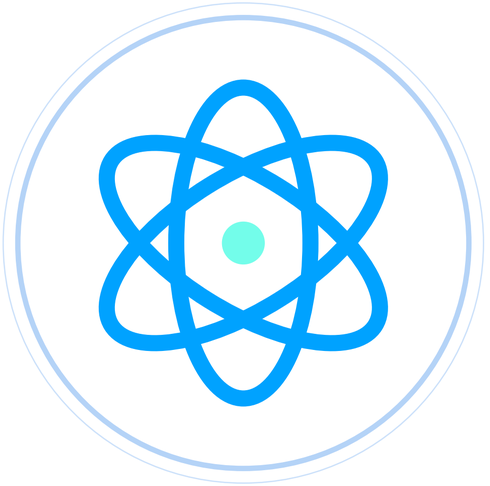
服务治理基础组件

服务注册与发现

Service registration and discovery

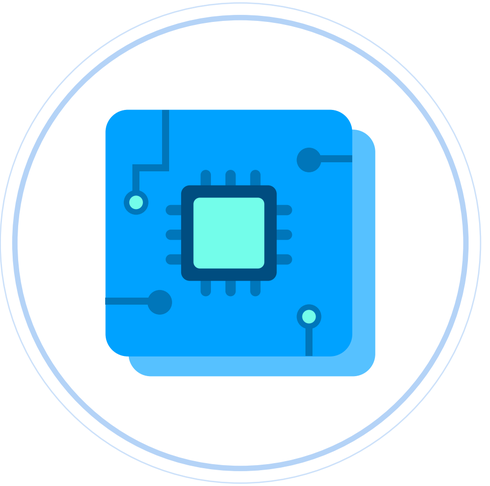
负载均衡

Load balancing



服务熔断

Circuit Breakers

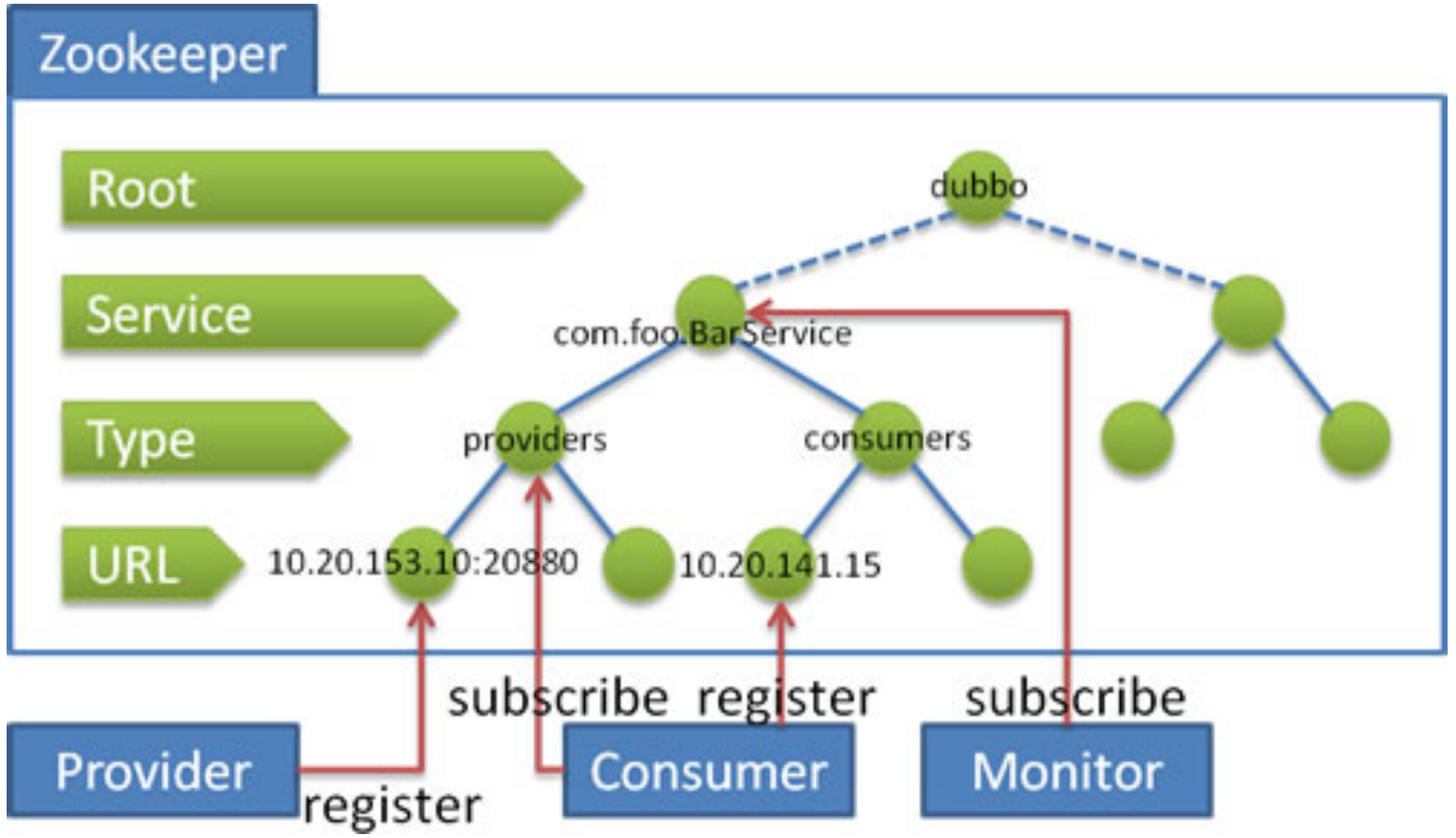


分布式配置

Distributed configuration

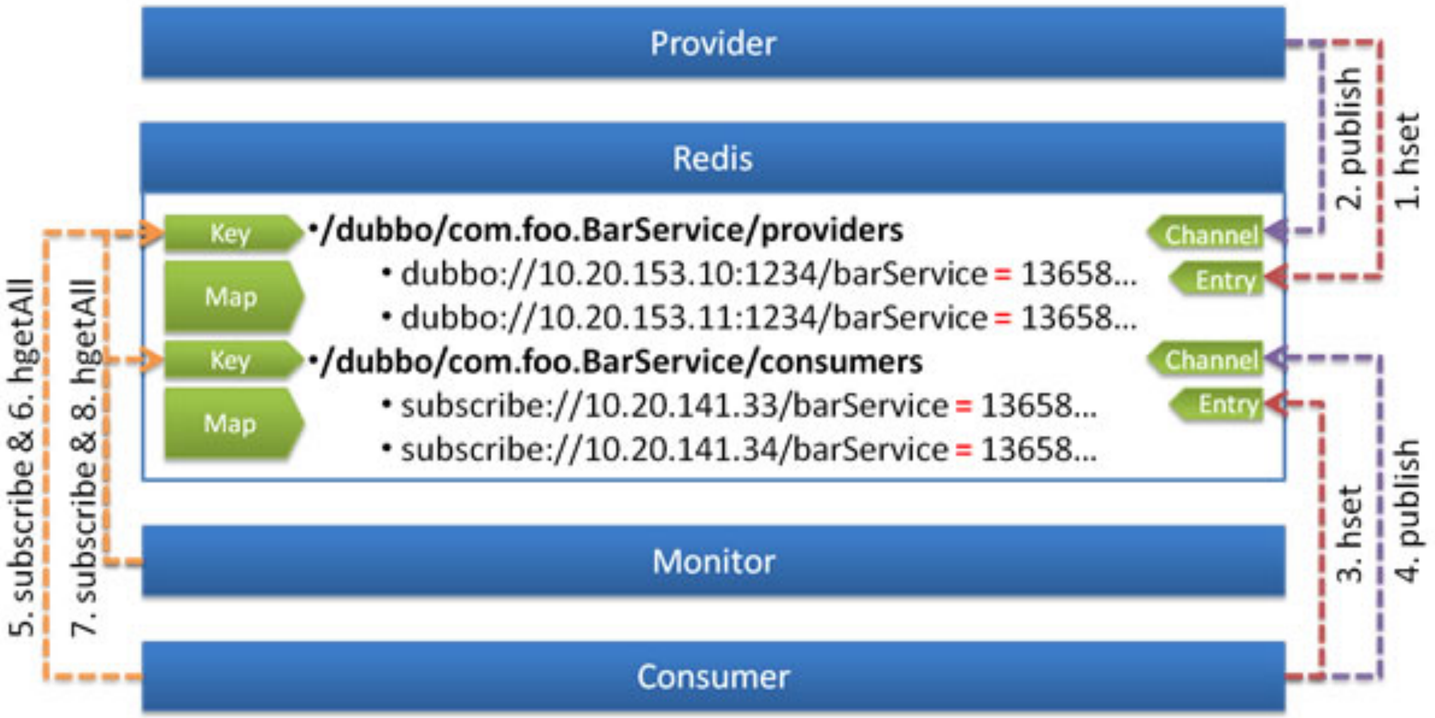
服务发现与注册


Apache
Zookeeper


Zookeeper
Root
Service
Type
URL
dubbo
com.foo.BarService
providers
consumers
10.20.153.10:20880
10.20.141.15
subscribe
register
subscribe
Provider
register
Consumer
Monitor

服务发现与注册

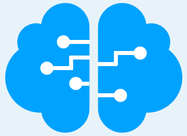

redis


Provider
Redis
Key
/dubbo/com.foo.BarService/providers
Map
dubbo://10.20.153.10:1234/barService = 13658...
dubbo://10.20.153.11:1234/barService = 13658...
Key
/dubbo/com.foo.BarService/consumers
Map
subscribe://10.20.141.33/barService = 13658...
subscribe://10.20.141.34/barService = 13658...
Channel
Entry
Channel
Entry
2. publish
1. hset
3. hset
4. publish
5. subscribe & 6. hgetAll
7. subscribe & 8. hgetAll
Monitor
Consumer

服务发现与注册

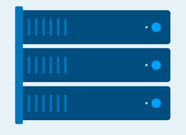

成熟协调系统
Dubbo、Spring Cloud
等适配方案
维护成本
客户端、Session 状态、
网络故障
CAP 理论
CP 模型，ZAB 算法，强
数据一致性
23
伸缩性限制
内存、GC，连接

负载均衡



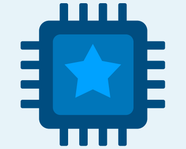
随机规则

com.alibaba.dubbo.rpc.cluster.loadbalance.RandomLoadBalance



轮训规则

com.alibaba.dubbo.rpc.cluster.loadbalance.RoundRobinLoadBalance



最不活跃规则

com.alibaba.dubbo.rpc.cluster.loadbalance.LeastActiveLoadBalance

服务熔断

Sentinel: Sentinel of Your Application

build passing | codecov 49% | Maven Central v1.3.0-GA | license Apache 2 | chat on gitter

What Does It Do?

As distributed systems become increasingly popular, the reliability and stability between services is becoming more important than ever before. Sentinel takes "flow" as breakthrough point, and works on multiple fields including **flow control**, **concurrency**, **circuit breaking** and **load protection**, to protect service reliability.

https://github.com/alibaba/Sentinel

Hystrix: Latency and Fault Tolerance for Distributed Systems

oss lifecycle maintenance | build passing | maven central 1.5.18 | License Apache 2

Hystrix Status

Hystrix is no longer in active development, and is currently in maintenance mode.

Hystrix (at version 1.5.18) is stable enough to meet the needs of Netflix for our existing applications. Meanwhile, our focus has shifted towards more adaptive implementations that react to an application's real time performance rather than pre-configured settings (for example, through adaptive concurrency limits). For the cases where something like Hystrix makes sense, we intend to continue using Hystrix for existing applications, and to leverage open and active projects like resilience4j for new internal projects. We are beginning to recommend others do the same.

https://github.com/Netflix/Hystrix

分布式配置

https://nacos.io/

Nacos 简介

Nacos 是 **Dynamic Naming and Configuration Service**的首字母简称，主要关注的领域是:

- 名字服务 (Naming Service)

名字服务提供分布式系统中所有对象(Object)、实体(Entity)的名字到关联的元数据之间的映射管理服务，例如 ServiceName -> Location Info, Distributed Lock Name -> Lock Owner/Status Info, 其中最主要的场景之一就是分布式系统中的服务发现，DNS系统就是一个典型的名字服务的实现。Nacos 会支持目前几乎所有常见微服务生态的服务发现，这包括 Kubernetes Service，Spring Cloud RESTful Service，Dubbo/gRPC RPC Service。

- 配置及服务元数据管理 (Configuration Service)

配置管理关注现代应用架构中服务及服务治理的元数据、规则以及应用的配置的管理，追求以中心化(centralized), 外部化 (externalized)以及动态化(Dynamic Style) 方式的存储与管理配置。相较与传统的基于分散的配置文件的应用配置管理方式，这种配置管理的方式更适应微服务架构。Kubernetes ConfigMap 以及 Spring Cloud Config Server 就是这种管理方式的一个实现。

Nacos 社区

- Nacos社区交流群（**已满**）
- Nacos社区交流群2（**已满**）
- Nacos社区交流群3（**已满**）
- Nacos社区交流群4（**已满**）
- Nacos社区交流群5（**还有机会**）

服务治理重新实现

Demo

- Case 1 ： Nacos 作为 Dubbo 注册中心
- Case 2 ： Nacos 作为 Dubbo 配置中心
- Case 3 ： Sentinel 作为 Dubbo 熔断实现
- Case 4 ： Nacos 作为 Sentinel 规则配置

Nacos 迁移方案

业务挑战：

如何将微服务集群从老注册中心平滑迁移到Nacos上?

技术限制：

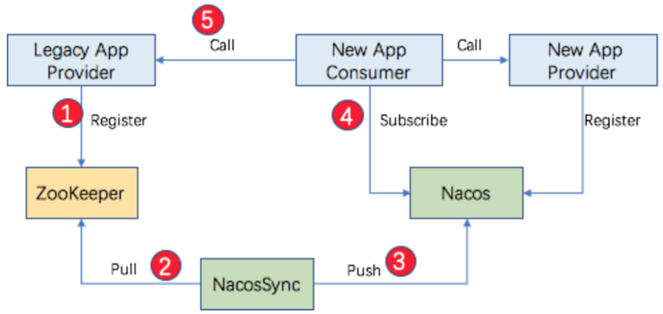
以下问题其实任何一个问题解了，都会简单很多。

- NacosSync的简单实现只支持 异构注册中心 -> Nacos的同步。(否则其他注册中心重复注册可能有问题，尚待验证)
- 应用在订阅的时候，只能订阅单服务注册中心，订阅多个注册中心很复杂。

Nacos 迁移方案

场景1：基于Nacos新应用上线，但是要调用老应用。

方案：直接使用NacosSync同步，如下图。



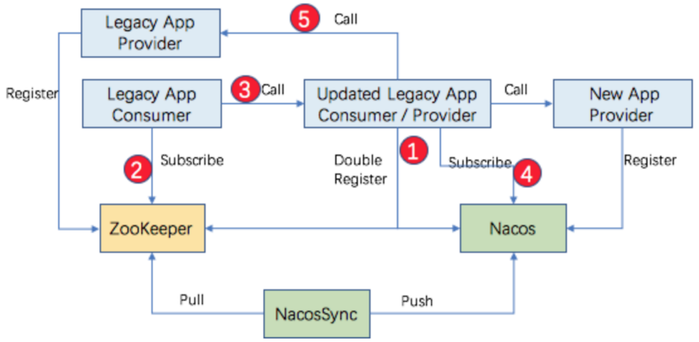
同步步骤：

1. 老应用Provider注册到老注册中心中(如ZK)
2. NacosSync接受到老注册中心的注册信息
3. NacosSync将注册信息同步到Nacos中
4. 新应用Consumer订阅Nacos应用信息
5. 新应用Consumer调用老应用Provider服务

Nacos 迁移方案

场景2：基于场景1，老应用A要调用新应用，但是同时又要被老应用B调用

方案：由于老应用要调用新应用，所以肯定是要改造。借助改造，将老应用应用改造为双注册即可。如下图



同步步骤(注，省略了一些前面的NacosSync同步信息)：

1. 老应用A改造后，双注册到Nacos和ZooKeeper
2. 老应用B通过通过ZooKeeper订阅到老应用A的注册信息
3. 老应用B调用改造的老应用A
4. 老应用A订阅到老应用C通过NacosSync同步过来的的注册信息
5. 老应用A调用老应用C

Nacos 社区

开放！Join US！持续贡献及一起发展社区

DISS is cheap, show me your hand!

有一种爱叫做放手！

向社区学习，向使用者学习！

“超哥”的微信，入群暗号“Nacos”

邮件组：nacos_dev@linux.alibaba.com

Thank you !