

目前市面上的开源产品

Disconf

2014 年 7 月百度开源的配置管理中心，同样具备配置的管理能力，目前已经不维护了，最近的一次代码提交是两年前的了。

Spring Cloud Config

2014 年 9 月开源，Spring Cloud 生态组件，与 Spring Cloud 体系无缝整合。

Apollo

2016 年 5 月，携程框架部开源的配置管理中心，具备规范的权限、流程治理等特性。

Nacos

2018 年 6 月，阿里开源的配置中心，可以做 DNS 和 RPC 的服务发现。

为什么选择 Apollo

社区活跃

刚刚发布了[1.4.0](#)版本，Issue 处理速度快

文档齐全

体验，部署，设计文档都齐全

Apollo 都有哪些重要功能

重要的灰度发布

想发布一台机器试试水，可以！

开源协议友好

Apache 2 license {{{TEASER_END}}}

Apollo 都有哪些重要功能

以下摘自[官网说明](#)

统一管理不同环境、不同集群的配置

配置修改实时生效（热发布）

版本发布管理

目前只支持对最近版本的恢复。<https://github.com/ctripcorp/apollo/issues/1642>

灰度发布

权限管理、发布审核、操作审计

编辑与发布是两个独立的操作。

Apollo 的组成

客户端配置信息监控

提供 **Java** 和 **.Net** 原生客户端

提供开放平台 **API**

这样可以通过其它方式来查看配置信息，[平台 API 说明](#)

使用方便的后台配置系统 **Portal**

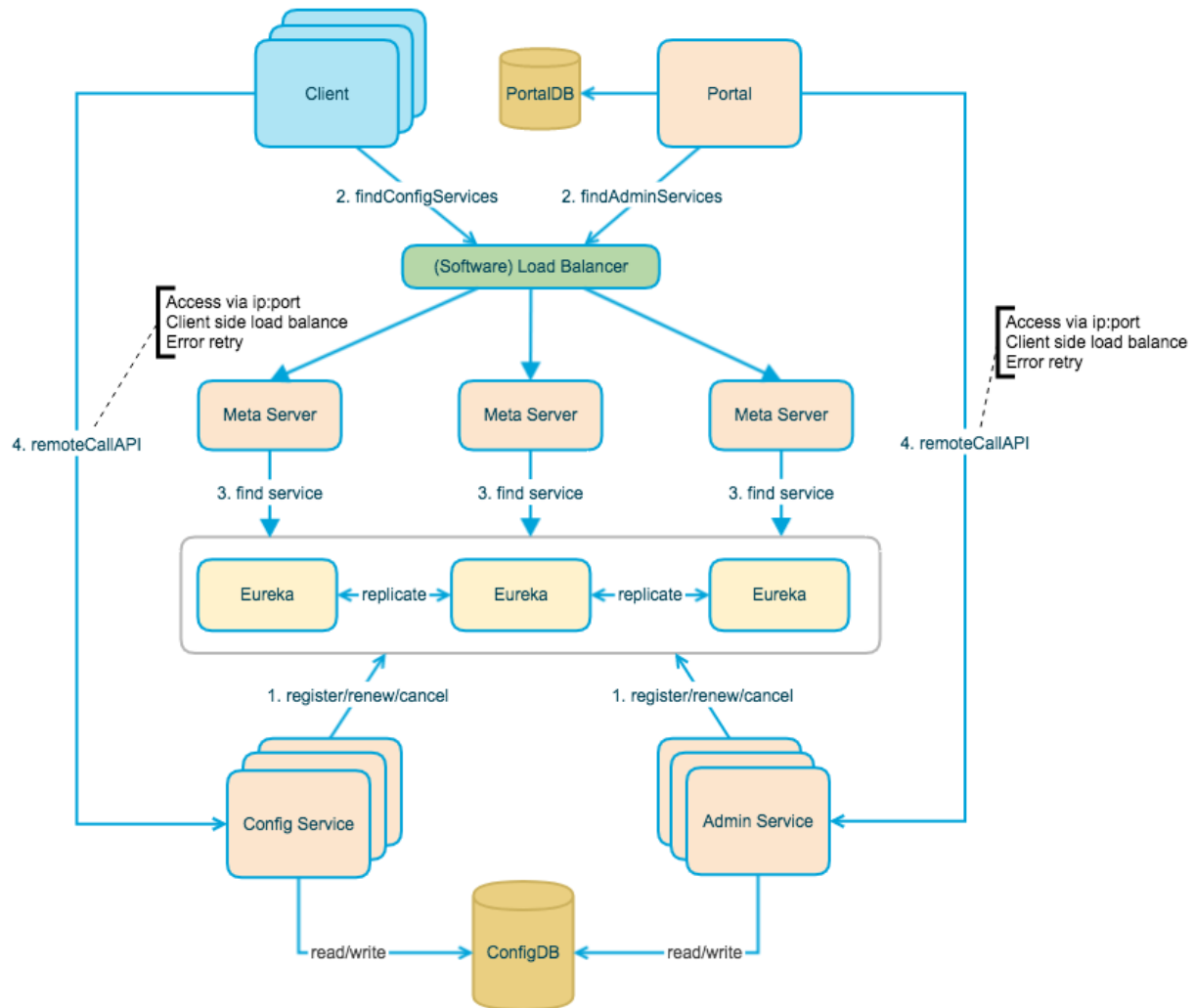
分布式部署相对较复杂，这是缺点

外部依赖少，目前依赖 Mysql

Apollo 的组成

Apollo 长什么样?

img-url:/images/apollo-home-screenshot.png



Apollo 整体设计

1. 总体架构

img-url:/images/apollo-overall-architecture.png

2. 代码结构

img-url:/images/apollo-code-structure.png

[各模块概要介绍](#)

Apollo Config Service

提供配置获取接口，服务对象为 Apollo 客户端

Apollo Admin Service

提供配置管理（修改、发布）接口，服务与 Portal

Apollo Portal

提供 WEB 界面供用户管理配置

Apollo 的重要设计

Admin Service 与 Config Service 的通信方式

Apollo 使用 Mysql 实现消息 (ReleaseMessage) 的处理，消息内容为 AppId+Cluster+Namespace，具体的设计思想可以参考[这里](#)

img-url:/images/apollo-release-message-design.png

客户端与服务端的通信方式

客户端与服务端保持一个长连接 (通过 Http Long Polling 实现) [Client Server](#)

重要的 Namespace

Namespace 是配置项的集合，类似于一个配置文件的概念，获取的权限分为 private 与 public 两种权限。

1. Namespace 的类型

体验部署方式

- 私有类型
- 公有类型
- 关联类型（继承类型）

Cluster 能用来做什么？

分机房实例，分任务功能实例，比如在一些实例执行 job，需要增加-Dapollo.cluster= 配置指定集群名

体验部署方式

Docker，本地源代码，虚拟机多环境

使用注意事项

- 在主版本有未发布的配置项时，要发布灰度后的全量则需要先把主版本中的配置发布后方可操作
- 使用 docker 运行 demo 测试时连接不上 docker 中的服务解决方法：<https://github.com/ctripcorp/apollo/issues/1481#issuecomment-422057411w>
- /opt 目录要有读写权限，日志文件默认打印在此目录
- Client 端要使用本地缓存配置，默认情况下确保/opt/data 目录存在
- 线上环境配置建议配置在机器上，不建议配置在代码中，与代码解耦
- config 与 admin service 要开放端口 8080 与 8090(默认情况下)，端口可以在项目中的 startup.sh 中修改
- 如果 Config Service 配置开启了内存缓存数据 (config-service.cache.enabled)，要提前考虑数据量的大小，调整服务的内存配置
- 在配置单环境 (如开发环境) 高可用时配置修改点
 - 修改 config 库的 eureka.service.url 配置项为多个开发环境的 meta server 地址
 - 修改 portal 的 apollo-env.properties 文件中的开发环境指定地址
 - 修改 admin 与 config 的连接数据库地址，都连接到开发环境的 mysql 地址

后续待确认问题处理

客户端的读取权限

目前看官方没有对这方面进行处理，维护者已将该功能加为 TODO。具体的实现方案看[这里](#)

docker 的实例获取问题

验证的是实例列表获取的是 docker 实例的 ip 地址，不同的 docker 机获取的 ip 是相同的

img-url:/images/apollo-docker-ip.png

本地调试读取配置问题

Apollo 的开发模式可以解决这个问题，具体看[这里](#)

Client 怎么来修改配置

目前看 Apollo-Client 包只提供从配置中心获取配置的功能，如果想对配置进行修改，则只能通过 OpenApi 的方式来实现。

配置指定时间生效

目前来看需要通过 OpenApi 实现

参考文档

<https://github.com/ctripcorp/apollo>

<https://github.com/alibaba/nacos>

<http://dockone.io/article/8767>