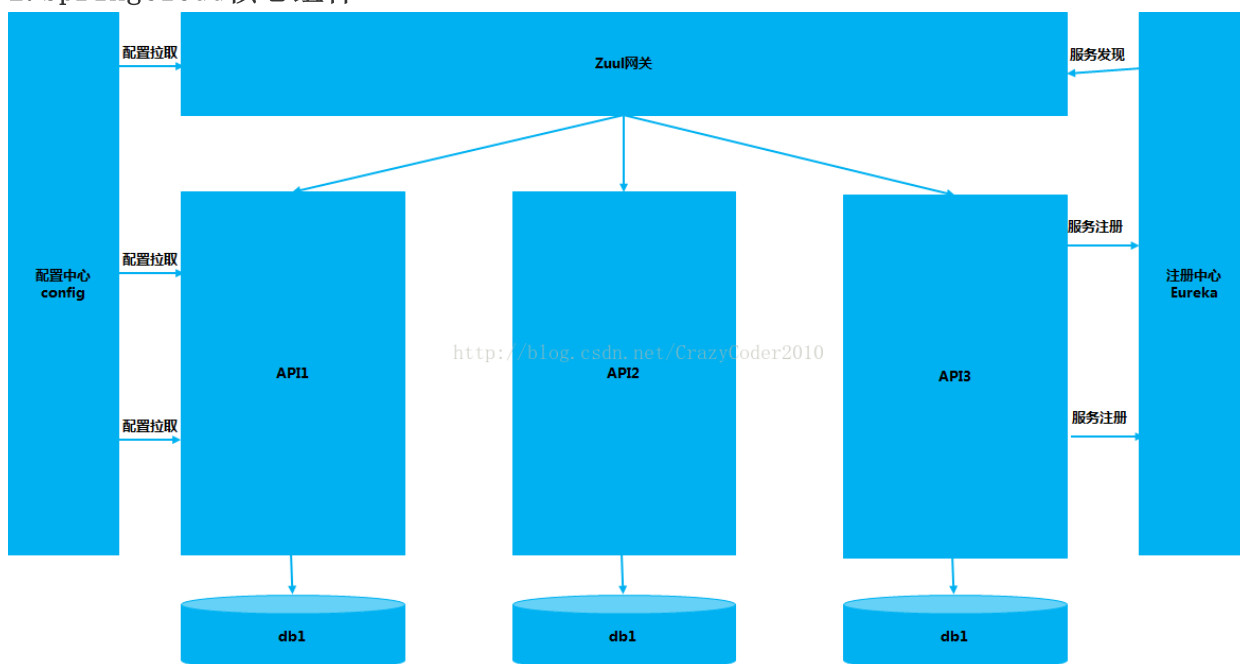


1. SpringCloud是什么

要解释SpringCloud的概念就必须先介绍一下微服务，微服务的英文名称成为Microservice，其特点是将整个web应用按照业务领域拆分成一系列的小的web服务，这些小的web服务可以被独立的编译打包和部署，并通过各自暴露的HTTP API进行相互通信，各个微服务之间相互配合，作为一个整体对外提供用户功能。微服务只是一种架构模式，其强调2点，第一点是‘小’，即每个服务的职责应该是单一的，一个微服务就是一个单一开发部署单元，第二点是微服务之间通过轻量级的HTTP API调用，彼此之间没有较重的耦合。

SpringCloud是基于SpringBoot实现的一套微服务框架，由Spring开源社区负责维护升级。它提供了实现微服务开发所必须的工具和组件，如配置管理、服务注册、断路器、服务网关等，另外SpringCloud基于SpringBoot开发实现，用SpringCloud来开发微服务会非常快捷。

2. SpringCloud核心组件



SpringCloud提供了一系列子项目

- **SpringCloud Config**

配置管理开发工具，可以把项目的配置信息集中放置一个远程服务器，如git/svn或其他类似的组件上，使用该组件后变更配置项就只要在一个集中的地方修改一次，各个微服务就可以主动获取最新配置信息（需要结合SpringCloud Bus之类的同步机制）

- **SpringCloud Zuul**

边缘服务工具，提供诸如服务路由，监控安全认证等功能。

- **SpringCloud Eureka**

服务注册中心，同时提供了服务端负载均衡功能

- **SpringCloud Consul**

封装了Consul的功能，Consul可以用作服务注册中心和配置中心使用

- **SpringCloud Sleuth**

日志收集工具，封装了Dapper, Zipkin和HTrace功能

- **SpringCloud Security**

安全工具包，为你的应用添加安全控制，也提供了Oauth2的服务端和客户端的封装

- **SpringCloud Bus**

事件、消息总线，用于集群（如配置变化）中传播状态变化使用

SpringCloud的特点：

- a) 约定大于配置
- b) 开箱即用，快速启动
- c) 组件齐全，且耦合度低
- d) 适合于各种环境

3. SpringCloud的同类产品

国内用的比较多的应该是阿里巴巴开源的dubbo框架了，如果要对dubbo和springcloud做个对比，我觉得springcloud有如下优势：

- **公司的背景**

阿里巴巴是一家全球商务公司，在国内知名度非常高，同时阿里系开源了一系列的开源软件，这些框架在国内的应用度还是非常广泛的。SpringCloud是Spring Source的产物，整合了一部分来自Netflix的产物，Spring Source在国外的影响力非常大，而且有Spring社区的背书让其成为Java业界最具影响力的组织了，Netflix开源的整套微服务架构套件是SpringCloud的核心。

- **社区的活跃度**

我们选择一个框架或者架构，其社区活跃度也是一个非常重要的考量因素，活跃度越高，解决问题的速度越快，框架也就会越来越完善，否则我们遇到问题时必须自己解决。Dubbo从2016年依旧基本就不再更新升级了（最近阿里刚又启动了Dubbo的后续升级工作），而SpringCloud的更新频率非常频繁，每天都有新的代码的代码改动在提交，由此可以看出，两者的活跃度时候不可同日而语的，这也可能与两家公司的性质相关，阿里巴巴是一家商业公司，将其部分组件或者框架开源可能只是他们回馈社会的一种方式，或者内部有开源的指标，一旦开源后并没有硬性的升级指标，而SpringCloud则是由Spring Source长期升级维护。

- **架构的完整度**

Dubbo只是实现了服务治理的一部分功能，如服务注册与发现，服务间调用，dubbo自带的控制台和监控中心也只是实现了简单的服务治理功能，有很多东西如服务的调用追踪等，如果选用了dubbo，这些东西不得不自行去实现或与其他框架做整合。而SpringCloud则是一套完整的微服务框架生态体系，提供了一种轻量级的，组件化的方式，让我们只要使用了SpringCloud，几乎不用手动去编写诸如分布式追踪等复杂的代码逻辑了，只要引入对应的组件，并按照文档的配置参数配置即可。

- **文档的质量**

Dubbo框架的文档质量可谓非常高了，也提供了中文和英文版本的，从框架设计到用法，非常细致深入，由于dubbo基本上很少升级，因此不同版本之间的差异很小。SpringCloud体系由于整合了大量的组件，因此代码的体量上要比dubbo大的多，而文档的内容大多数是以组件之间的整合使用为主

- **通信协议的差异**

Dubbo框架默认使用的是自行实现的dubbo协议进行进程之间的调用，是基于tcp协议的，因此dubbo服务之间的调用效率非常高，而SpringCloud则采用HTTP协议作为微服务之间

的通信标准，个人认为，随着现在硬件的发展，HTTP协议调用带来的空间和效率的浪费其实可能会越来越不明显；另外用了dubbo，服务的调用者和消费者之间是一种强依赖的关系——代码层面的依赖（dto），而HTTP协议是就文本的，也就是一纸协议，双方只要按照协议实现接口，并不需要做代码层面的依赖；第三点就是dubbo只能做基于Java语言之间的进程通信，如果涉及到异构系统之间的数据传输，dubbo就不擅长了（当当开源了个dubbox插件，个人认为还是比较蹩脚），而SpringCloud以HTTP作为协议就注定了它的通用性，很容易与遗留系统或异构系统进行对接整合。