Spring Cloud的核心成员、以及架构实现详细介绍

题目标签

学习时长：20分钟

题目难度：中等

知识点标签：Spring Cloud

题目描述

Spring Cloud的核心成员、以及架构实现详细介绍

1. 面试题分析

根据题目要求我们可以知道：

1.什么是微服务

2.SOA和微服务的区别

3.微服务架构优势

4.什么是Spring Boot

5.什么是Spring Cloud

6.Spring Cloud的核心成员

7.Spring Cloud架构实现

8.微服务、Spring Cloud、Spring Boot三者关系

> 分析需要全面并且有深度

容易被忽略的坑

- 分析片面
- 没有深入

1.什么是微服务

微服务的概念源于Martin Fowler所写的一篇文章“Microservices”。

微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，服务之间互相协调、互相配合，为用户提供最终价值。每个服务运行在其独立的进程中，服务与服务间采用轻量级的通信机制互相沟通（通常是基于HTTP的RESTful
API）。每个服务都围绕着具体业务进行构建，并且能够被独立地部署到生产环境、类生产环境等。另外，应尽量避免统一的、集中式的服务管理机制，对具体的一个服务而言，应根据业务上下文，选择合适的语言、工具对其进行构建。

微服务是一种架构风格，一个大型复杂软件应用由一个或多个微服务组成。系统中的各个微服务可被独立部署，各个微服务之间是松耦合的。每个微服务仅关注于完成一件任务并很好地完成该任务。在所有情况下，每个任务代表着一个小的业务能力。

2.SOA和微服务的区别

Martin Fowler提出来这一概念可以说把SOA的理念继续升华，精进了一步。 微服务架构强调的第一个重点就是**业务系统需要彻底的组件化和服务化**，原有的单个业务系统会拆分为多个可以独立开发，设计，运行和运维的小应用。这些小应用之间通过服务完成交互和集成。

从服务粒度上，既然是微，必然微服务更倡导服务的细粒度，重用组合，甚至是每个操作（或方法）都是独立开发的服务，足够小到不能再进行拆分。而SOA没有这么极致的要求，只需要接口契约的规范化，内部实现可以更粗粒度，微服务更多为了可扩充性、负载均衡以及提高吞吐量而去分解应用，但同时也引发了打破数据模型以及维护一致性的问题。

从部署方式上，这个是最大的不同，对比以往的Java
EE部署架构，通过展现层打包WARs，业务层划分到JARs最后部署为EAR一个大包，而微服务则把应用拆分成为一个一个的单个服务，应用Docker技术，不依赖任何服务器和数据模型，是一个
全栈应用，可以通过自动化方式独立部署，每个服务运行在自己的进程。

如果一句话来谈SOA和微服务的区别，即微服务不再强调传统SOA架构里面比较重的ESB企业服务总线，同时SOA的思想进入到单个业务系统内部实现真正的组件化。

3.微服务架构优势

1.粒度更细(可维护和效率)

在将应用分解，每一个微服务专注于单一功能，并通过定义良好的接口清晰表述服务边界。由于体积小、复杂度低，每个微服务可由一个小规模开发团队完全掌控，易于保持高可维护性和开发效率。

2.独立部署

由于微服务具备独立的运行进程，所以每个微服务也可以独立部署。

3.容错

在微服务架构下，故障会被隔离在单个服务中。若设计良好，其他服务可通过重试、平稳退化等机制实现应用层面的容错。

4.扩展

单块架构应用也可以实现横向扩展，就是将整个应用完整的复制到不同的节点。

4.什么是Spring Boot

Spring
Boot 框架是由 Pivotal 团队提供的全新框架,其设计目的是用来简化基于 Spring
应用的初始搭建以及开发过程。SpringBoot
框架使用了特定的方式来进行应用系统的配置,从而使开发人员不再需要耗费大量精力去定义模板化的配置文件。

5.什么是Spring Cloud

3. Spring Cloud 是一个基于 Spring Boot 实现的云应用开发工具,它为基于 JVM 的云应用开发中的配置管理、服务发现、断路器、智能路由、微代理、控制总线、全局锁、决策竞选、分布式会话和集群状态管理等,是微服务的一种实现。

6.Spring Cloud的核心成员

1.Spring Cloud Netflix

Spring Cloud Netflix 集成众多Netflix的开源软件：Eureka, Hystrix, Zuul, Archaius，组成了微服务的最重要的核心组件。

2.Netflix Eureka

服务中心，用于服务注册与发现，一个基于 REST 的服务，用于定位服务。

3.Netflix Hystrix

熔断器，容错管理工具，旨在通过熔断机制控制服务和第三方库的节点,从而对延迟和故障提供更强大的容错能力。

4.Netflix Zuul

Zuul 是在云平台上提供动态路由,监控,弹性,安全等边缘服务的框架。

5.Netflix Archaius

配置管理API，包含一系列配置管理API，提供动态类型化属性、线程安全配置操作、轮询框架、回调机制等功能，可以实现动态获取配置。

6.Spring Cloud Config

配置中心，利用git集中管理程序的配置。

7.Spring Cloud Bus

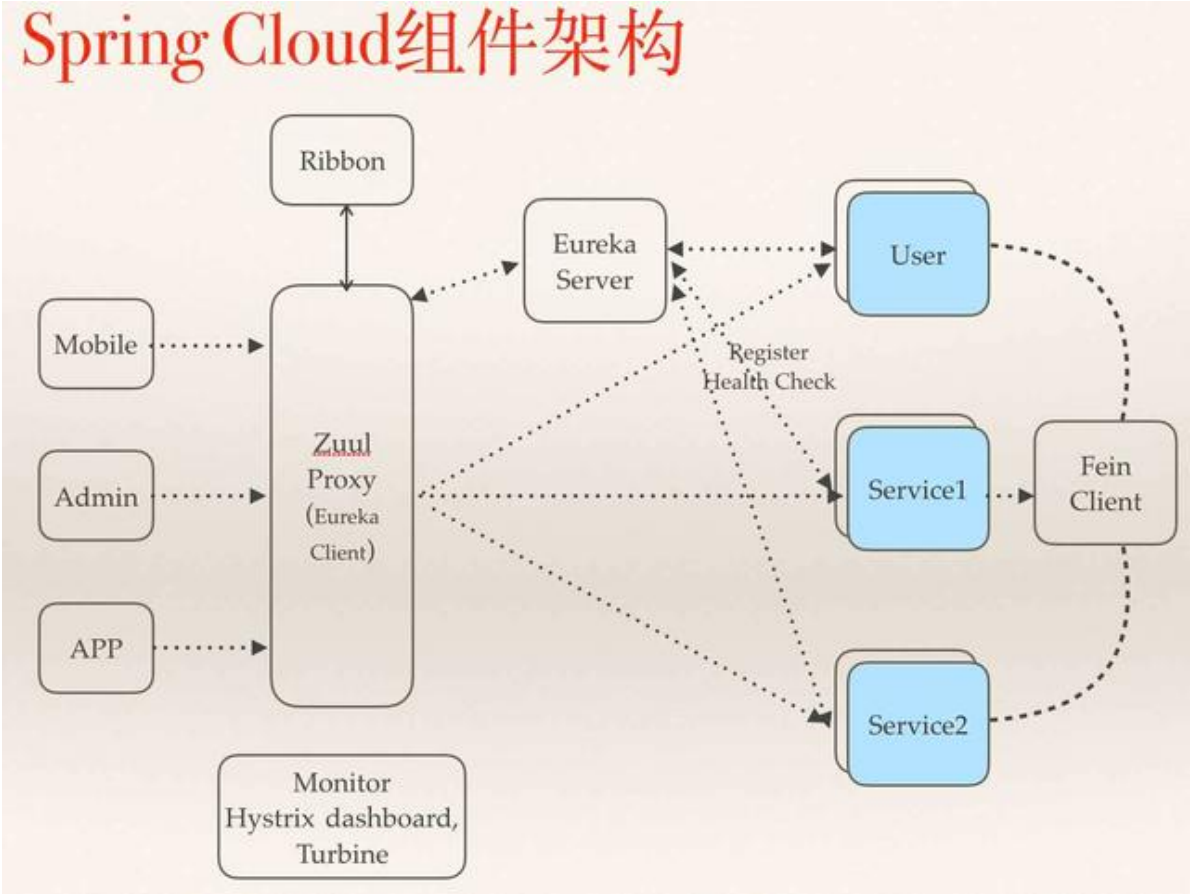
事件、消息总线，用于在集群（例如，配置变化事件）中传播状态变化，可与Spring Cloud Config联合实现热部署。

8.Spring Cloud Ribbon

Ribbon是Netflix发布的负载均衡器，它有助于控制HTTP和TCP的客户端的行为。为Ribbon配置服务提供者地址后，Ribbon就可基于某种负载均衡算法，自动地帮助服务消费者去请求。

7.Spring Cloud架构实现

通过这张图，可以比较清楚的了解到各组件配置使用运行机制：



1、请求统一通过API网关（Zuul）来访问内部服务.

2、网关接收到请求后，从注册中心（Eureka）获取可用服务

3、由Ribbon进行均衡负载后，分发到后端具体实例

4、微服务之间通过Feign进行通信处理业务

5、Hystrix负责处理服务超时熔断

6、Turbine监控服务间的调用和熔断相关指标

微服务、Spring Cloud、Spring Boot三者关系

微服务是一种架构的理念，提出了微服务的设计原则，从理论为具体的技术落地提供了指导思想。

1. SpringBoot专注于快速方便的开发单个个体微服务。
2. SpringCloud是关注全局的微服务协调整理治理框架，它将SpringBoot开发的一个个单体微服务整合并管理起来，
3. 为各个服务之间提供，配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、精选决策、分布式会话等集成服务。
4. SpringBoot可以离开SpringCloud独立开发项目，但是SpringCloud离不开SpringBoot，属于依赖关系。
5. SpringBoot专注于快速、方便的开发单个微服务个体，SpringCloud关注全局的服务治理框架。

2. 扩展内容

- Spring Cloud与Dubbo的详细比较
- 从单体架构、到SOA、再到微服务的架构设计详解
- 微服务Dubbo和SpringCloud架构设计、优劣势比较
- Dubbo与SpringCloud的Ribbon、Hystrix、Feign的优劣势比较