博客来源于个人对分享会的理解,如有错误欢迎指正.

背景

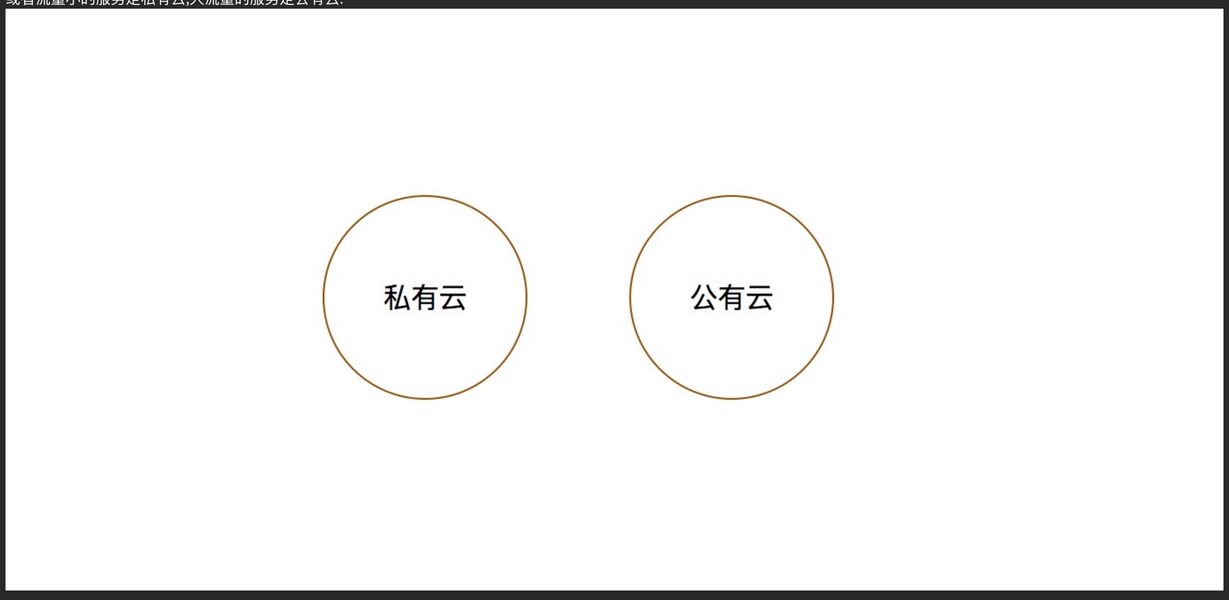
私有云微服务系统在大促的时候会扩容机器,那么在大促之前就需要采购物理机.这就造成了大促之后机器闲置,资源浪费.另外,机器准备是需要时间的,大促进行中如果发现准备不充分就尴尬了.

混合云

为了解决这个问题,利用阿里云、腾讯云之类的共有云的资源按需分配易扩展的特性,形成混合云的技术方案.

在平常的时候只部署私有云,在大促的时候,将服务部署到公有云上面.

或者流量小的服务走私有云,大流量的服务走公有云.



有了两套云服务,就可以在接入层根据url流量来分配服务调度.

服务化框架的单元化路由

使用这个方案要注意,每个接口所依赖的微服务需要完整的部署在相应的云上面.

比如图中的create接口和query接口.

我们假设create接口依赖A、B、C、D四个微服务,query接口依赖C、D、E三个微服务.那么在私有云上面要部署完整的A、B、C、D微服务,在公有云上面也需要部署C、D、E微服务.如果不这么做,即使公有云和私有云之间的服务调用的打通的,也会有效率问题.

假设create和query接口需要依赖新的微服务F,而F只在私有云上部署,那么公有云上面的query接口就会有问题.这里可以引入注册中心,在公有云上没有F时,调用私有云上面的服务,并提出告警,直到公有云上面部署F为止.

混合云架构图如下:

同城双活

随着公司业务的发展,流量越来越大,集群规模也越来越大,可能就会有将服务部署在不同集群的需求.服务部署在同一集群的时候,如果机房发生火灾地震之类灾难,就会对公司造成不可挽回的损失.

基于以上需求,和混合云的技术策略.我们就可以将服务部署在不同的云机房.

用户维度流量调度

不但可以将单元服务部署在不同机房,还可以将同一个单元服务部署在不同机房.我们希望同一个用户多次访问同样服务的时候,可以只访问一个固定的机房.

这就LB除了能根据url不同路由到不同机房,还要能根据其他条件,如用户id来路由服务.

job和服务耦合问题

在单服务单机部署转换到单服务多机部署之前,会存在一些历史遗留问题.比如将job任务和api服务写到同一个工程里面,这会造成一些麻烦.

如上图,应用部署之后,job就会在三台机器上都有.在同一时间,会执行3次相同的任务,这会造成资源浪费,有时候也会产生错误.

首先要做的是将定时任务和api服务分离.然后是使用统一job调度平台.

这会方便任务统计、分配、管理.想象一下你有成千上万个不同的定时任务分散在不同的代码实现中,其中有些是spring schedue实现;有些是使用线程轮训实现;有些cron表达式硬编码在代码中;有些在数据库某张表里面.这已经很头大了,如果有些定时任务是在linux里面写shell脚本,通过crontab调用,在维护时想死的心都会有.

统一调度平台有很多,比如:elastic-job、xxl-job等.

使用统一配置中心

想象一个场景,在部署新集群时,假设我们使用新的只读数据库,应用里的配置就应该指向这个新的数据库.如果没有配置中心,要么就得让开发人员新增一个专用配置文件,要么就要运维修改启动参数.不管是哪一方维护,这都是一个灾难.

这时我们需要引入配置中心,常用的配置中心有spring config、apollo等.

全链路系统诊断

我们想要知道一个请求在微服务中那个步骤出问题了,通过在全链路使用唯一id,具体如下:

1. 每次请求产生唯一traceid,产生唯一id的地方包括接口,网关,job调度等
2. RPC调用上下文透传traceid
3. mq消息属性透传traceid
4. traceid落库(mysql,redis等)

有了traceid,在配合ELK就可以很方便的查出服务在哪出的问题.

统一网关、mq容灾略

康彬-基于微服务的混合云和同城双活实践