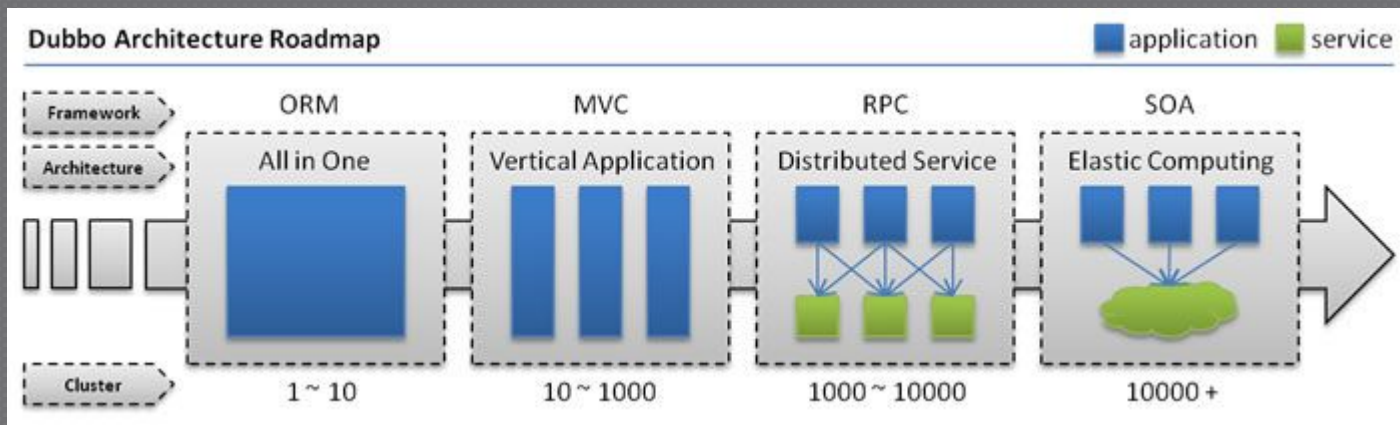




Dubbo 介绍

Dubbo 为 IPP 解决了什么，
不只是 IPP

Dubbo



单一应用架构

当网站流量很小时，只需一个应用，将所有功能都部署在一起，以减少部署节点和成本。

此时，用于简化增删改查工作量的数据访问框架(ORM)是关键。

垂直应用架构

当访问量逐渐增大，单一应用增加机器带来的加速度越来越小，将应用拆成互不相干的几个应用，以提升效率。

此时，用于加速前端页面开发的Web框架(MVC)是关键。

分布式服务架构

当垂直应用越来越多，应用之间交互不可避免，将核心业务抽取出来，作为独立的服务，逐渐形成稳定的服务中心，使前端应用能更快速的响应多变的市场需求。

此时，用于提高业务复用及整合的分布式服务框架(RPC)是关键。

流动计算架构

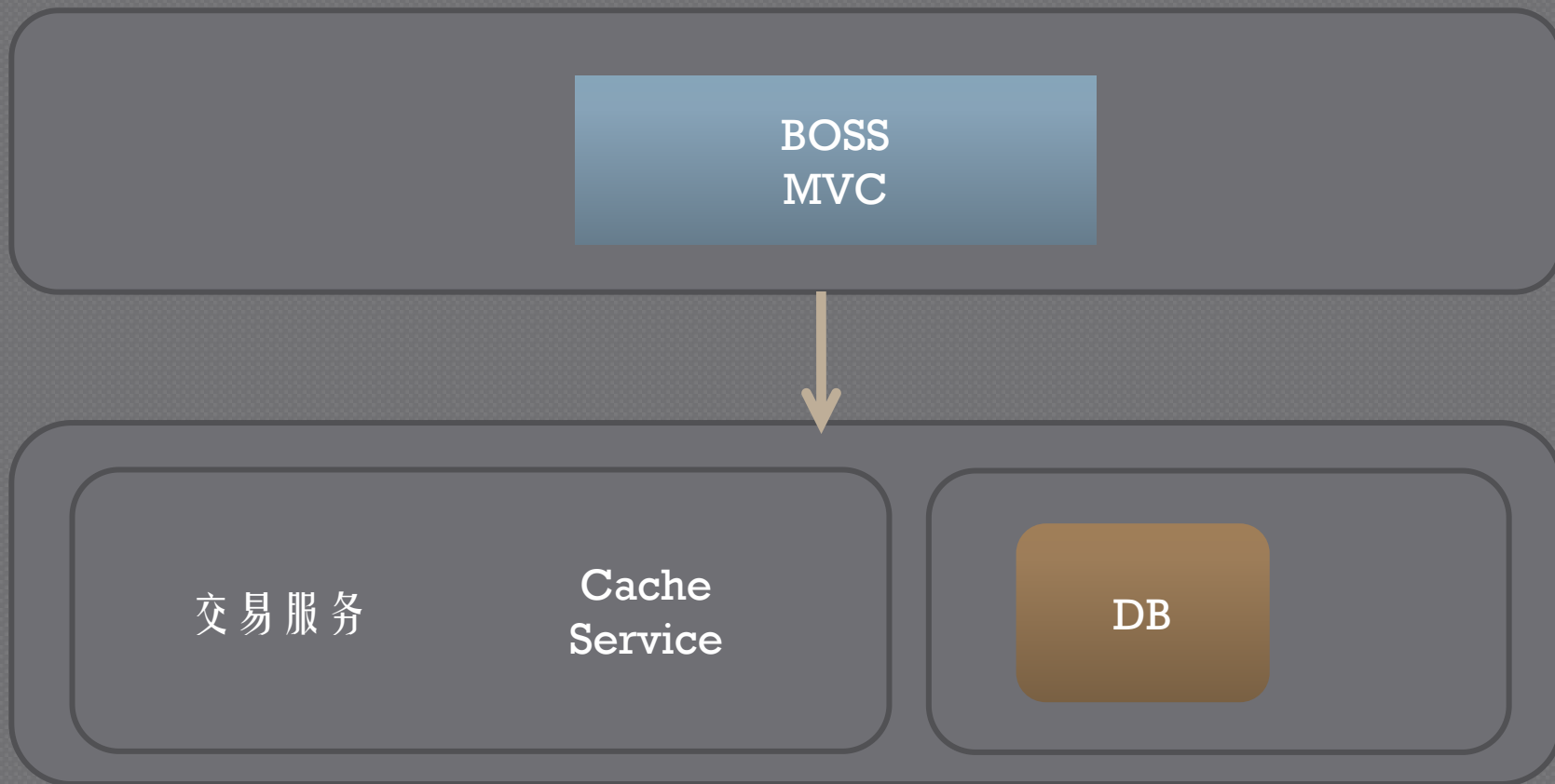
当服务越来越多，容量的评估，小服务资源的浪费等问题逐渐显现，此时需增加一个调度中心基于访问压力实时管理集群容量，提高集群利用率。

此时，用于提高机器利用率的资源调度和治理中心(SOA)是关键。

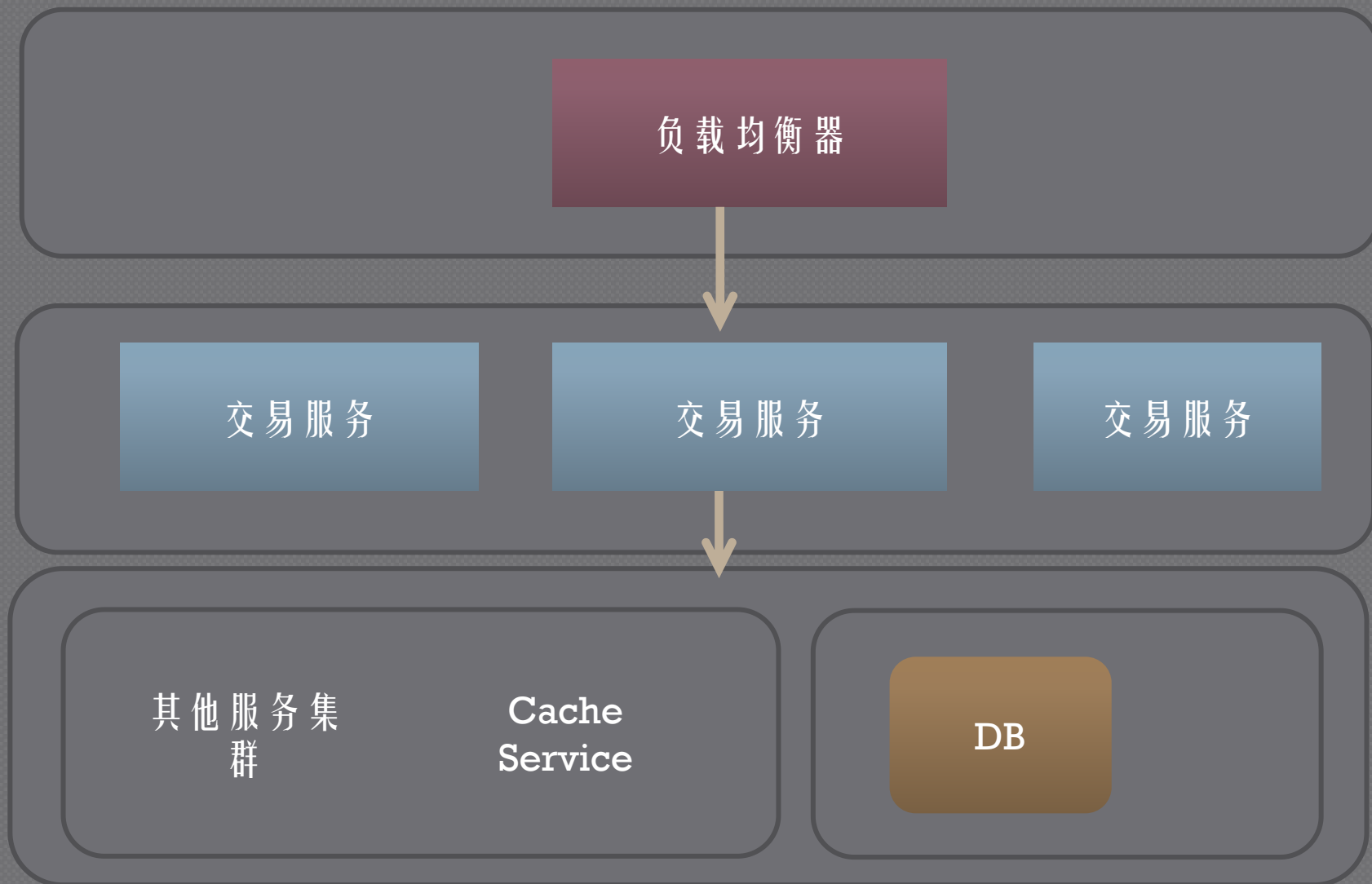
问题

我们现在在哪？
我们的目标在哪？

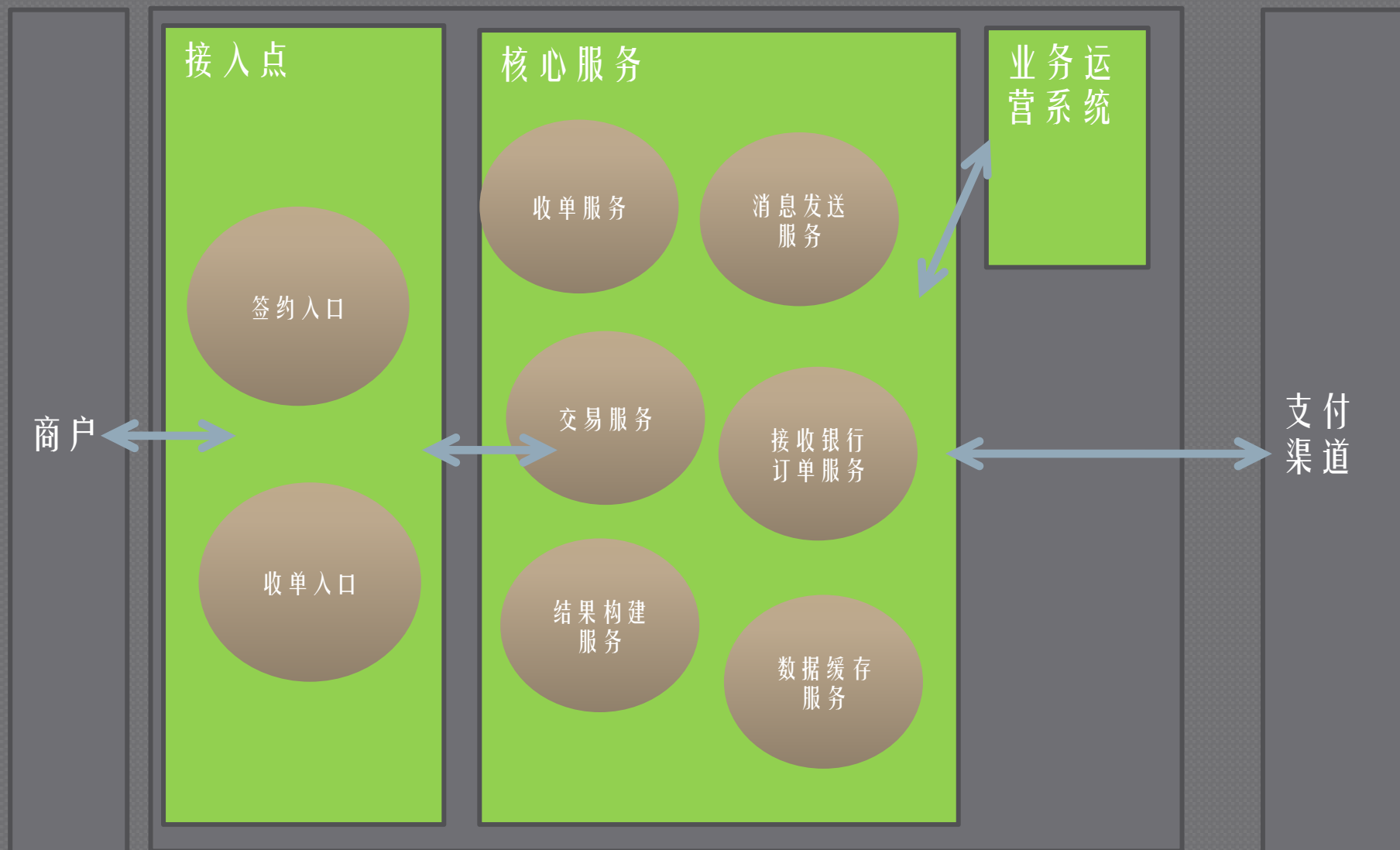
BOSS MVC 构架



交易服务集群



IPP 服务

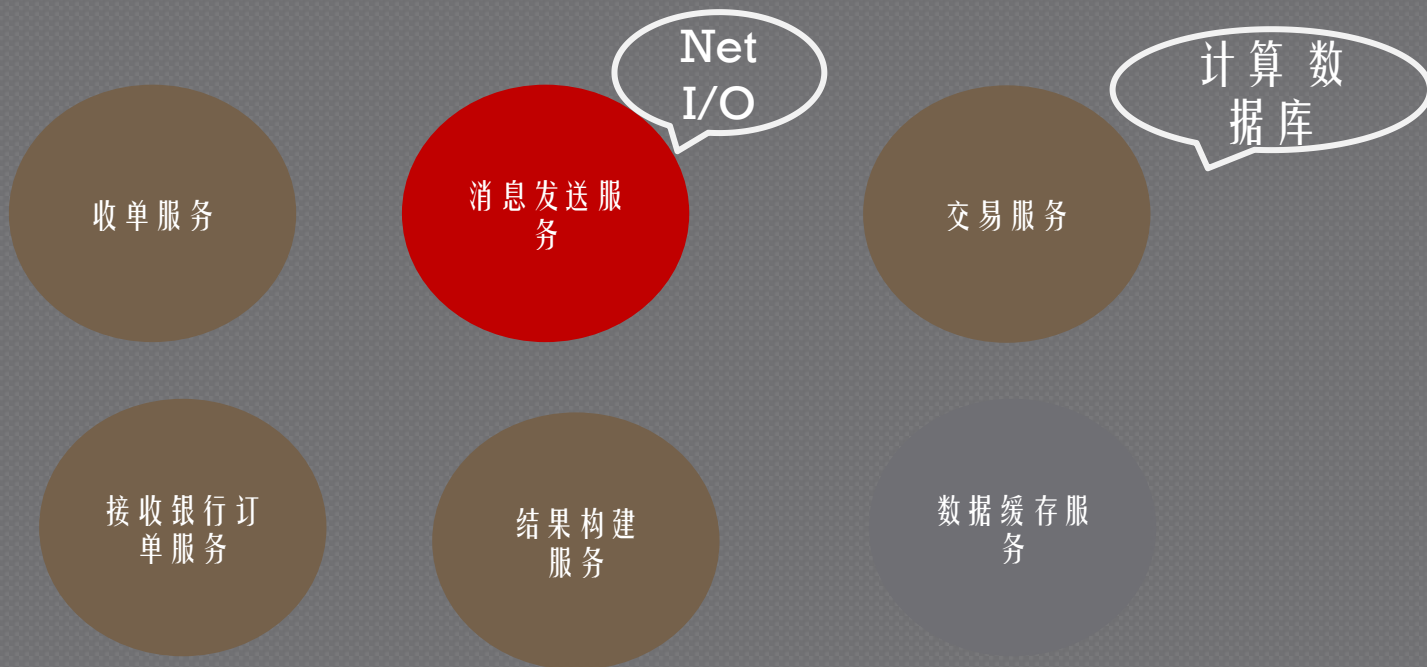


答案

IPP已经进入阶段三
但是金融平台没进入阶段三

更高级的阶段三，而不是入门级别

IPP 服务特点



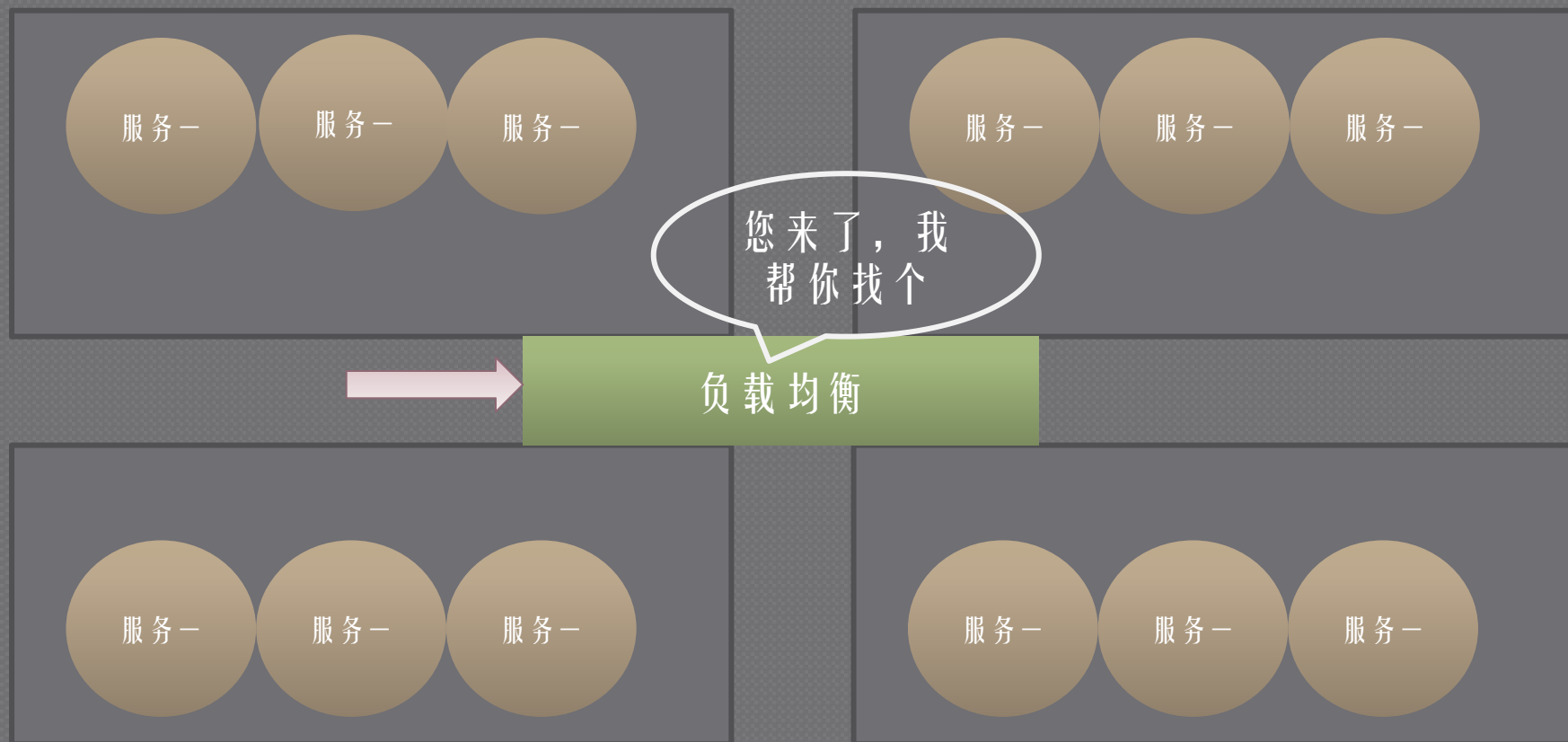
特点：

1. 服务基于hessian调用
2. 依赖于负载均衡的IP
3. 基于URL的负载
4. 通过F5进行负载（容易单点）

问题：

1. 依赖负载均衡
2. 服务无法动态扩展（计算型，I/O型）
3. 服务缺乏监控，不清楚服务状态
4. 版本升级

服务寻址



单点（压力，可靠性）
复杂的配置

大规模之前

在大规模服务化之前，应用可能只是通过RMI或Hessian等工具，简单的暴露和引用远程服务，通过配置服务的URL地址进行调用，通过F5等硬件进行负载均衡。

(1) 当服务越来越多时，服务URL配置管理变得非常困难，F5硬件负载均衡器的单点压力也越来越大。

此时需要一个服务注册中心，动态的注册和发现服务，使服务的位置透明。

并通过在消费方获取服务提供方地址列表，实现软负载均衡和Failover，降低对F5硬件负载均衡器的依赖，也能减少部分成本。

(2) 当进一步发展，服务间依赖关系变得错综复杂，甚至分不清哪个应用要在哪个应用之前启动，架构师都不能完整的描述应用的架构关系。

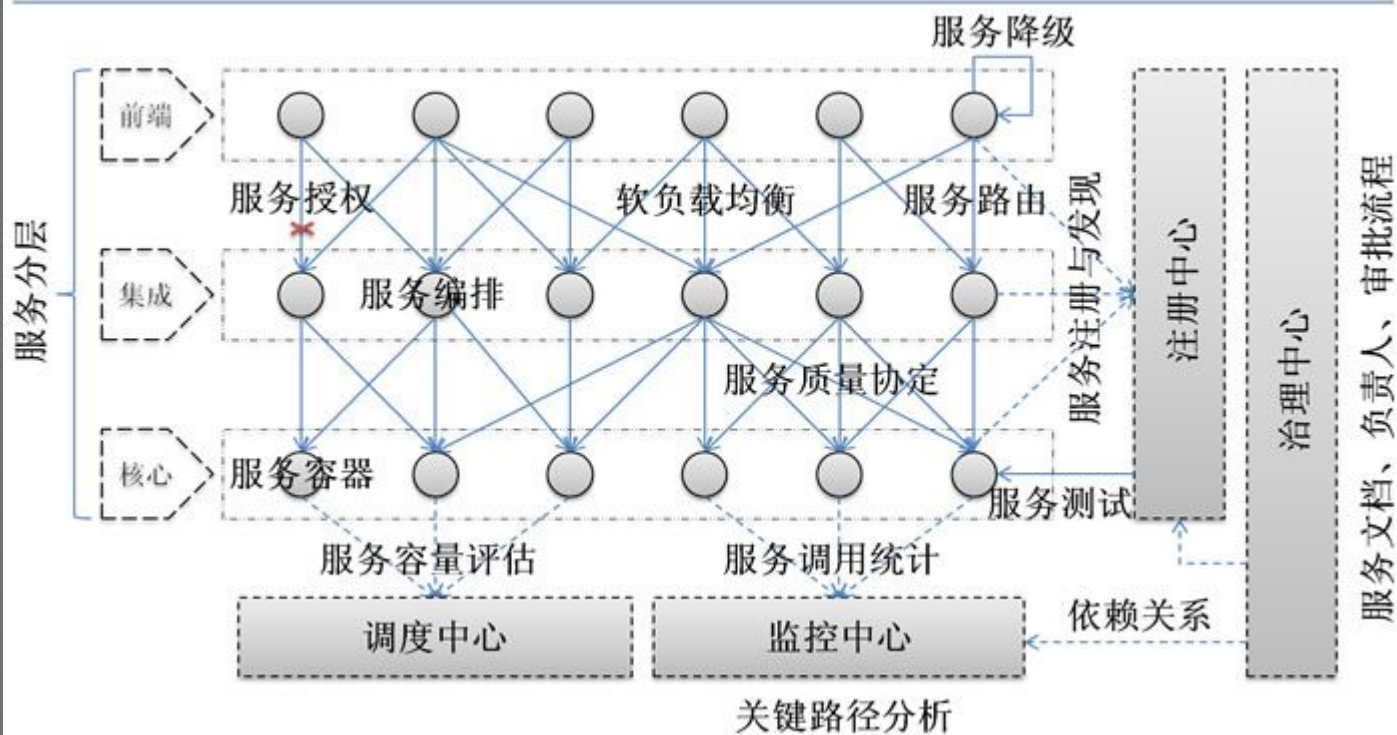
这时，需要自动画出应用间的依赖关系图，以帮助架构师理清关系。

(3) 接着，服务的调用量越来越大，服务的容量问题就暴露出来，这个服务需要多少机器支撑？什么时候该加机器？

为了解决这些问题，第一步，要将服务现在每天的调用量，响应时间，都统计出来，作为容量规划的参考指标。

其次，要可以动态调整权重，在线上，将某台机器的权重一直加大，并在加大的过程中记录响应时间的变化，直到响应时间到达阈值，记录此时的访问量，再以此访问量乘以机器数反推总容量。

Dubbo服务治理



Dubbo 为我们带来了什么

以前：

服务上线

安装服务，配置负责均衡

服务下线

配置负责均衡，代价大

服务升级

较好的方式：url 添加版本，

然后配置负载均衡

路由选择

压力监控

无

引入 dubbo 之后：

服务上线

安装个服务就可以了，不需要配置

服务下线

监控中心取消注册

服务升级

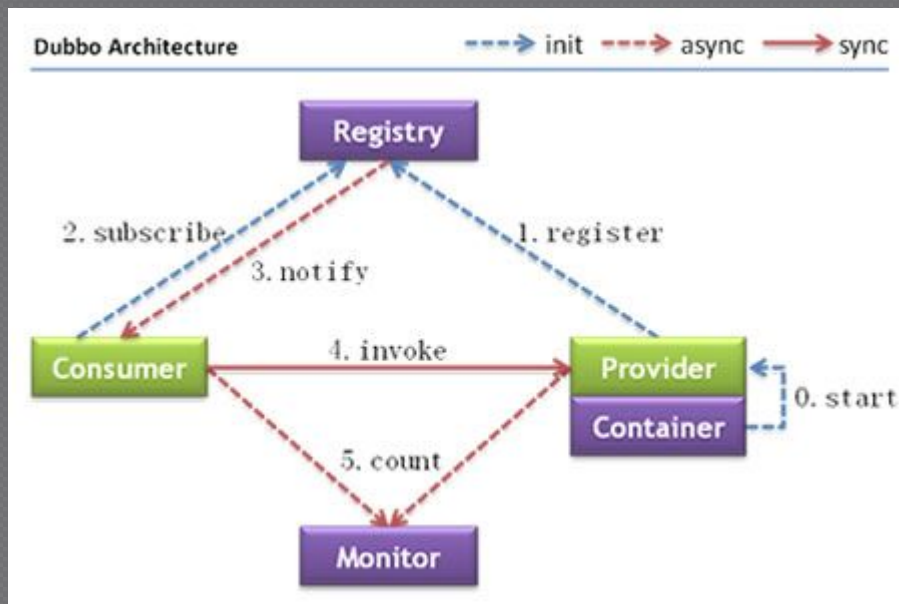
配置引用

路由选择

待研究

压力监控

Dubbo 构架



Provider: 暴露服务的服务提供方。

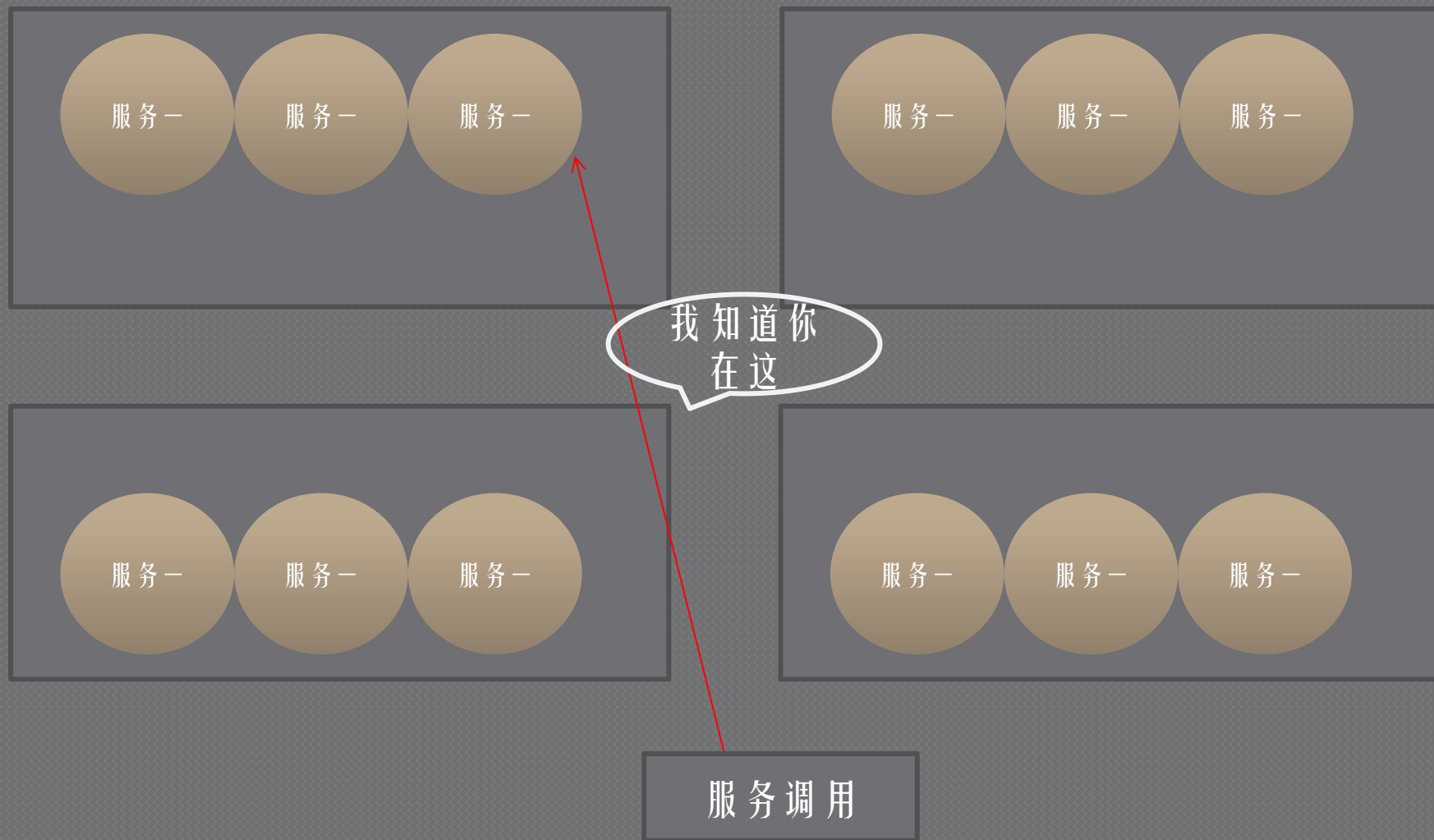
Consumer: 调用远程服务的服务消费方。

Registry: 服务注册与发现的注册中心。

Monitor: 统计服务的调用次数和调用时间的监控中心。

Container: 服务运行容器。

新的构架



监控

[Home](#)[Applications](#)[Services](#)[Hosts](#)[Registries](#)[Servers](#)[Status](#)[Log](#)[System](#)[Home](#) > [Applications](#)

Applications (7)

Application Name: <input type="text"/>	Owner	Providers(1)	Consumers(0)	Depends On(4)	Used By(4)
simple-monitor		Providers(1)	No consumer	No dependency	No used
TransactionWebsite		No provider	No consumer	Depends On(1)	No used
PaymentGatewayApp		No provider	No consumer	Depends On(1)	Used By(3)
BOSS-IPP		No provider	No consumer	Depends On(1)	No used
ClearingServiceApp		No provider	No consumer	No dependency	No used
MessageDispatcher		No provider	No consumer	No dependency	Used By(1)
PaymentWebSite		No provider	No consumer	Depends On(1)	No used

监控

最多 接入方接口测试-页面... 通联理财平台-接入方... 通联管理系统 Dashboard [Jenkins] dev环境 dev_boss uat_通联理财平台-测... Dubbo uat_boss_通联管理系统

Home	Applications	Services	Hosts	Registries	Servers	Status	Log	System
Home > Applications > PaymentGatewayApp > Providers Consumers Depends On Used By								
Providers (26)								
Provider URL: <input type="text"/>								Unregister
hessian://192.168.11.163:8080/ipp-payment-app/com.allinpay.cpg.IPaymentQueryService?anyhost=true&application=PaymentGatewayApp&dubbo=2.5.4-SNAPSHOT&generic=false&interface=com.allinpay.cpg.IPaymentQueryService&methods=doQuery&pid=24680&revision=1.0.1&server=servlet&side=provider×tamp=1396503996252								Unregister
hessian://192.168.11.163:8080/ipp-payment-app/com.allinpay.cpg.cache.IMerchantWebsiteNameCacheHandler?anyhost=true&application=PaymentGatewayApp&dubbo=2.5.4-SNAPSHOT&generic=false&interface=com.allinpay.cpg.cache.IMerchantWebsiteNameCacheHandler&methods=generateCacheKey,loadDataIntoCache,removeData,refreshData,getKeyPrefix,setCacheManager,getData,initLoadDataIntoCache&pid=24680&revision=1.0.1&server=servlet&side=provider×tamp=1396503987987								Unregister
hessian://192.168.11.163:8080/ipp-payment-app/com.allinpay.cpg.cache.IMchtBizzCacheHandler?anyhost=true&application=PaymentGatewayApp&dubbo=2.5.4-SNAPSHOT&generic=false&interface=com.allinpay.cpg.cache.IMchtBizzCacheHandler&methods=generateCacheKey,loadDataIntoCache,removeData,refreshData,getKeyPrefix,setCacheManager,getData,initLoadDataIntoCache&pid=24680&revision=1.0.1&server=servlet&side=provider×tamp=1396503989328								Unregister
hessian://192.168.11.163:8080/ipp-payment-app/com.allinpay.cpg.IRecvPaymentRequestService?anyhost=true&application=PaymentGatewayApp&dubbo=2.5.4-SNAPSHOT&generic=false&interface=com.allinpay.cpg.IRecvPaymentRequestService&methods=processPaymentRequest&pid=24680&revision=1.0.1&server=servlet&side=provider×tamp=1396503994442								Unregister
hessian://192.168.11.163:8080/ipp-payment-app/com.allinpay.cpg.cache.IMerchantIssuerCacheHandler?anyhost=true&application=PaymentGatewayApp&dubbo=2.5.4-SNAPSHOT&generic=false&interface=com.allinpay.cpg.cache.IMerchantIssuerCacheHandler&methods=generateCacheKey,loadDataIntoCache,removeData,refreshData,getKeyPrefix,setCacheManager,getData,initLoadDataIntoCache&pid=24680&revision=1.0.1&server=servlet&side=provider×tamp=1396503987325								Unregister

使用方法（使用服务）

通过Spring配置引用远程服务：

consumer.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

    <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->
    <dubbo:application name="consumer-of-helloworld-app" />

    <!-- 使用multicast广播注册中心暴露发现服务地址 -->
    <dubbo:registry address="multicast://224.5.6.7:1234" />

    <!-- 生成远程服务代理，可以和本地bean一样使用demoService -->
    <dubbo:reference id="demoService" interface="com.alibaba.dubbo.demo.DemoService" />

</beans>
```

使用方法（暴露服务）

provider.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

  <!-- 提供方应用信息，用于计算依赖关系 -->
  <dubbo:application name="hello-world-app" />

  <!-- 使用multicast广播注册中心暴露服务地址 -->
  <dubbo:registry address="multicast://224.5.6.7:1234" />

  <!-- 用dubbo协议在20880端口暴露服务 -->
  <dubbo:protocol name="dubbo" port="20880" />

  <!-- 声明需要暴露的服务接口 -->
  <dubbo:service interface="com.alibaba.dubbo.demo.DemoService" ref="demoService" />

  <!-- 和本地bean一样实现服务 -->
  <bean id="demoService" class="com.alibaba.dubbo.demo.provider.DemoServiceImpl" />

</beans>
```