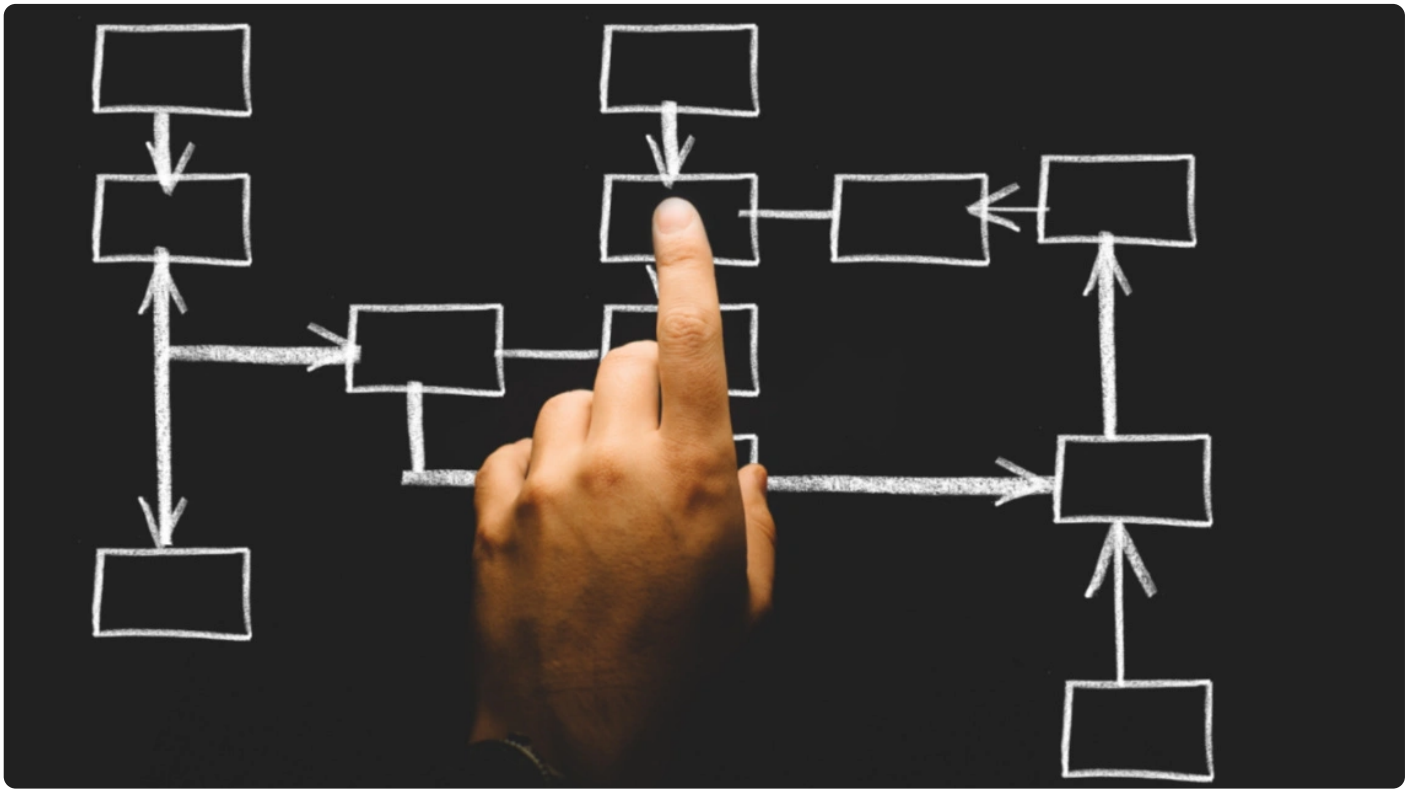
43 | 互联网架构模板：“用户层”和“业务层”技术

2018-08-04 李运华 来自北京

《从0开始学架构》



上一期，我从计算机网络层的角度谈了应对“高性能”和“高可用”的整体架构设计。今天，我将从“用户层”和“业务层”的角度谈谈常见的应用场景和关键技术。

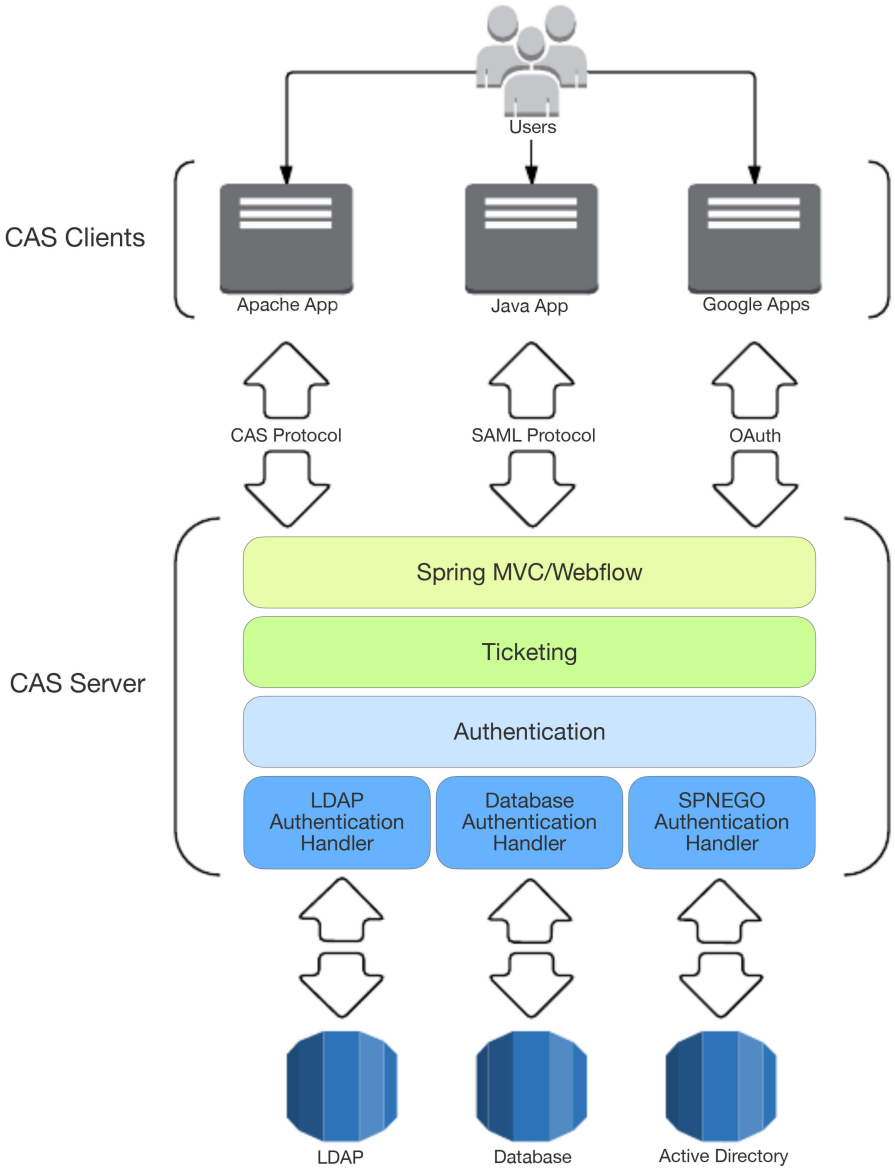
用户层技术

1. 用户管理

互联网业务的一个典型特征就是通过互联网将众多分散的用户连接起来，因此用户管理是互联网业务必不可少的一部分。

稍微大一点的互联网业务，肯定会涉及多个子系统，这些子系统不可能每个都管理这么庞大的用户，由此引申出用户管理的第一个目标：**单点登录（SSO）**，又叫统一登录。单点登录的技术实现手段较多，例如 cookie、JSONP、token 等，目前最成熟的开源单点登录方案当属

CAS，其架构如下（https://apereo.github.io/cas/4.2.x/planning/Architecture.html）：

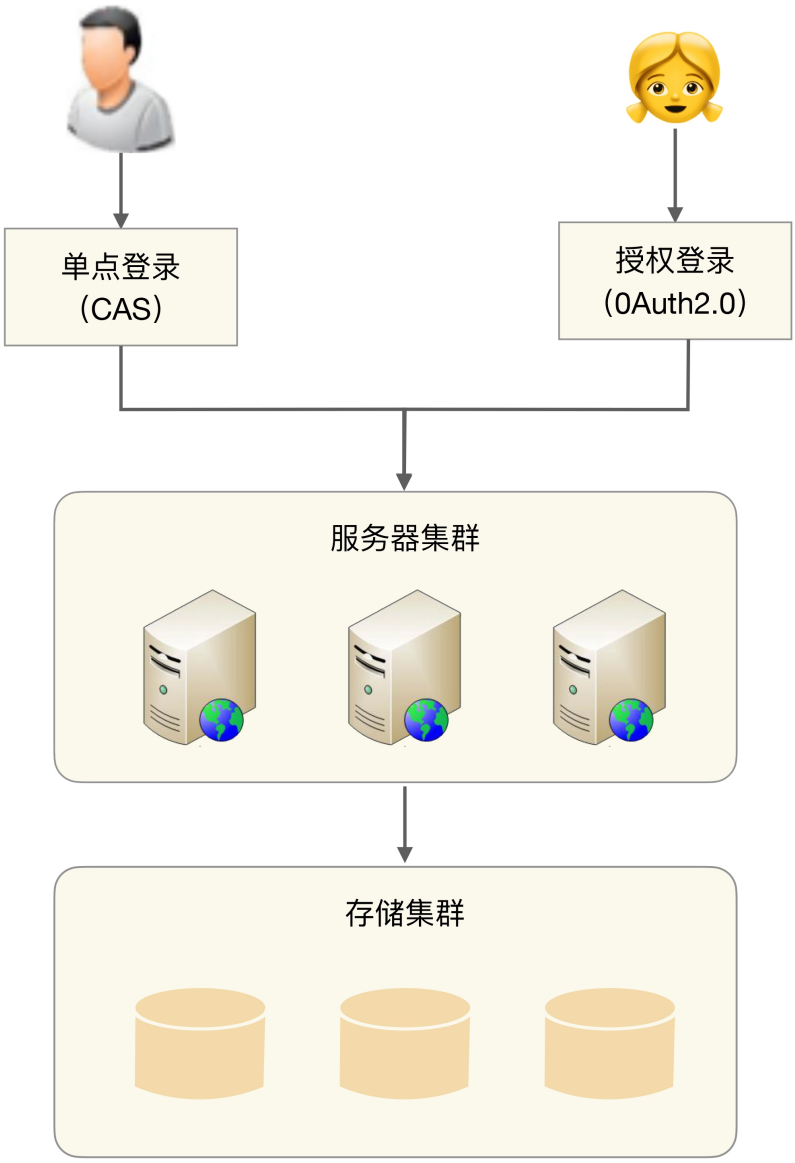


除此之外，当业务做大成为了平台后，开放成为了促进业务进一步发展的手段，需要允许第三方应用接入，由此引申出用户管理的第二个目标：**授权登录**。现在最流行的授权登录就是 OAuth 2.0 协议，基本上已经成为了事实上的标准，如果要做开放平台，则最好用这个协议，私有协议漏洞多，第三方接入也麻烦。

用户管理系统面临的主要问题是用户数巨大，一般至少千万级，QQ、微信、支付宝这种巨无霸应用都是亿级用户。不过也不要被这个数据给吓倒了，用户管理虽然数据量巨大，但实现起来并不难，原因是什么呢？ 因为用户数据量虽然大，但是不同用户之间没有太强的业务关

联，A 用户登录和 B 用户登录基本没有关系。因此虽然数据量巨大，但我们用一个简单的负载均衡架构就能轻松应对。

用户管理的基本架构如下：



2. 消息推送

消息推送根据不同的途径，分为短信、邮件、站内信、App 推送。除了 App，不同的途径基本上调用不同的 API 即可完成，技术上没有什么难度。例如，短信需要依赖运营商的短信接口，邮件需要依赖邮件服务商的邮件接口，站内信是系统提供的消息通知功能。

App 目前主要分为 iOS 和 Android 推送，iOS 系统比较规范和封闭，基本上只能使用苹果的 APNS；但 Android 就不一样了，在国外，用 GCM 和 APNS 差别不大；但是在国内，情况就复杂多了：首先是 GCM 不能用；其次是各个手机厂商都有自己的定制的 Android，消息推送实现也不完全一样。因此 Android 的消息推送就五花八门了，大部分有实力的大厂，都会自己实现一套消息推送机制，例如阿里云移动推送、腾讯信鸽推送、百度云推送；也有第三方公司提供商业推送服务，例如友盟推送、极光推送等。

通常情况下，对于中小公司，如果不涉及敏感数据，Android 系统上推荐使用第三方推送服务，因为毕竟是专业做推送服务的，消息到达率是有一定保证的。

如果涉及敏感数据，需要自己实现消息推送，这时就有一定的技术挑战了。消息推送主要包含 3 个功能：设备管理（唯一标识、注册、注销）、连接管理和消息管理，技术上面临的主要挑战有：

海量设备和用户管理
消息推送的设备数量众多，存储和管理这些设备是比较复杂的；同时，为了针对不同用户进行不同的业务推广，还需要收集用户的一些信息，简单来说就是将用户和设备关联起来，需要提取用户特征对用户进行分类或者打标签等。

连接保活
要想推送消息必须有连接通道，但是应用又不可能一直在前台运行，大部分设备为了省电省流量等原因都会限制应用后台运行，限制应用后台运行后连接通道可能就被中断了，导致消息无法及时的送达。连接保活是整个消息推送设计中细节和黑科技最多的地方，例如应用互相拉起、找手机厂商开白名单等。

消息管理
实际业务运营过程中，并不是每个消息都需要发送给每个用户，而是可能根据用户的特征，选择一些用户进行消息推送。由于用户特征变化很大，各种排列组合都有可能，将消息推送给哪些用户这部分的逻辑要设计得非常灵活，才能支撑花样繁多的业务需求，具体的设计方案可以采取规则引擎之类的微内核架构技术。

3. 存储云、图片云

互联网业务场景中，用户会上传多种类型的文件数据，例如微信用户发朋友圈时上传图片，微博用户发微博时上传图片、视频，优酷用户上传视频，淘宝卖家上传商品图片等，这些文件具备几个典型特点：

数据量大：用户基数大，用户上传行为频繁，例如 2016 年的时候微信朋友圈每天上传图片就达到了 10 亿张（http://mi.techweb.com.cn/tmt/2016-05-25/2338330.shtml）。

文件体积小：大部分图片是几百 KB 到几 MB，短视频播放时间也是在几分钟内。

访问有时效性：大部分文件是刚上传的时候访问最多，随着时间的推移访问量越来越小。

为了满足用户的文件上传和存储需求，需要对用户提供文件存储和访问功能，这里就需要用到前面我在专栏第 40 期介绍“存储层”技术时提到的“小文件存储”技术。简单来说，存储云和图片云通常的实现都是“CDN + 小文件存储”，现在有了“云”之后，除非 BAT 级别，一般不建议自己再重复造轮子了，直接买云服务可能是最快也是最经济的方式。

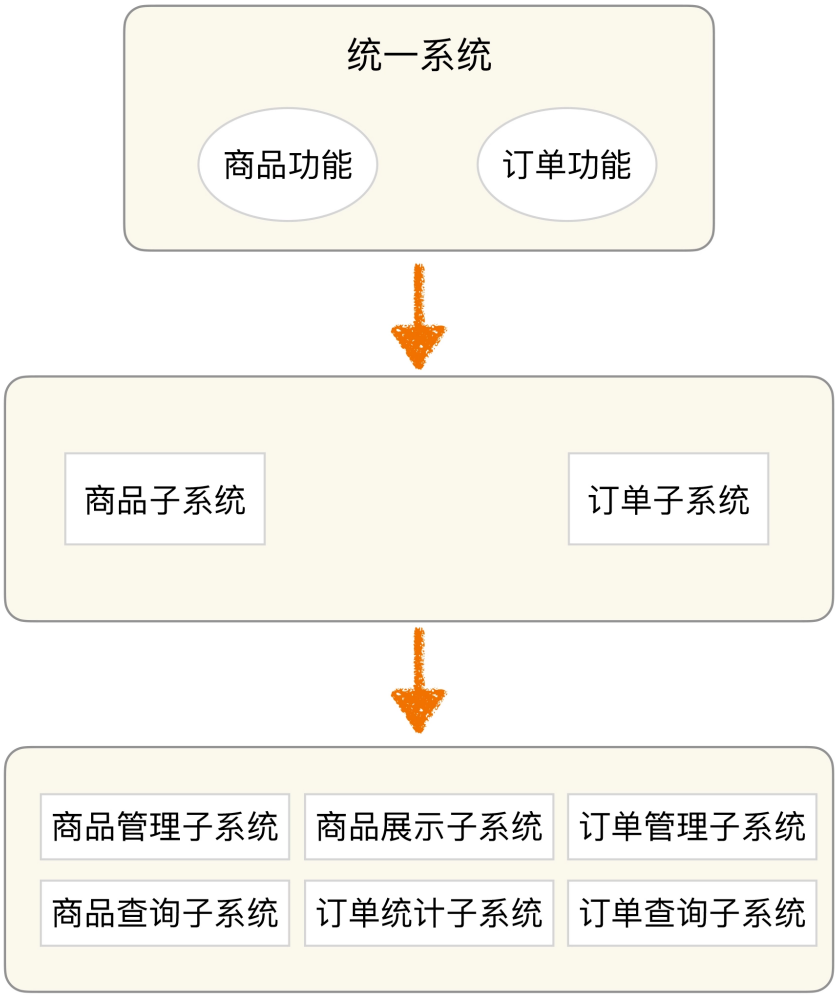
既然存储云和图片云都是基于“CDN + 小文件存储”的技术，为何不统一一套系统，而将其拆分为两个系统呢？这是因为“图片”业务的复杂性导致的，普通的文件基本上提供存储和访问就够了，而图片涉及的业务会更多，包括裁剪、压缩、美化、审核、水印等处理，因此通常情况下图片云会拆分为独立的系统对用户提供服务。

业务层技术

互联网的业务千差万别，不同的业务分解下来有不同的系统，所以业务层没有办法提炼一些公共的系统或者组件。抛开业务上的差异，各个互联网业务发展最终面临的问题都是类似的：业务复杂度越来越高。也就是说，业务层面对的主要技术挑战是“复杂度”。

复杂度越来越高的一个主要原因就是系统越来越庞大，业务越来越多。幸运的是，面对业务层的技术挑战，我们有一把“屠龙宝刀”，不管什么业务难题，用上“屠龙宝刀”问题都能迎刃而解。这把“屠龙宝刀”就是“拆”，化整为零、分而治之，将整体复杂性分散到多个子业务或者子系统里面去。具体拆的方式你可以查看专栏前面可扩展架构模式部分的分层架构、微服务、微内核等。

我以一个简单的电商系统为例，如下图所示：



我这个模拟的电商系统经历了 3 个发展阶段：

第一阶段：所有功能都在 1 个系统里面。

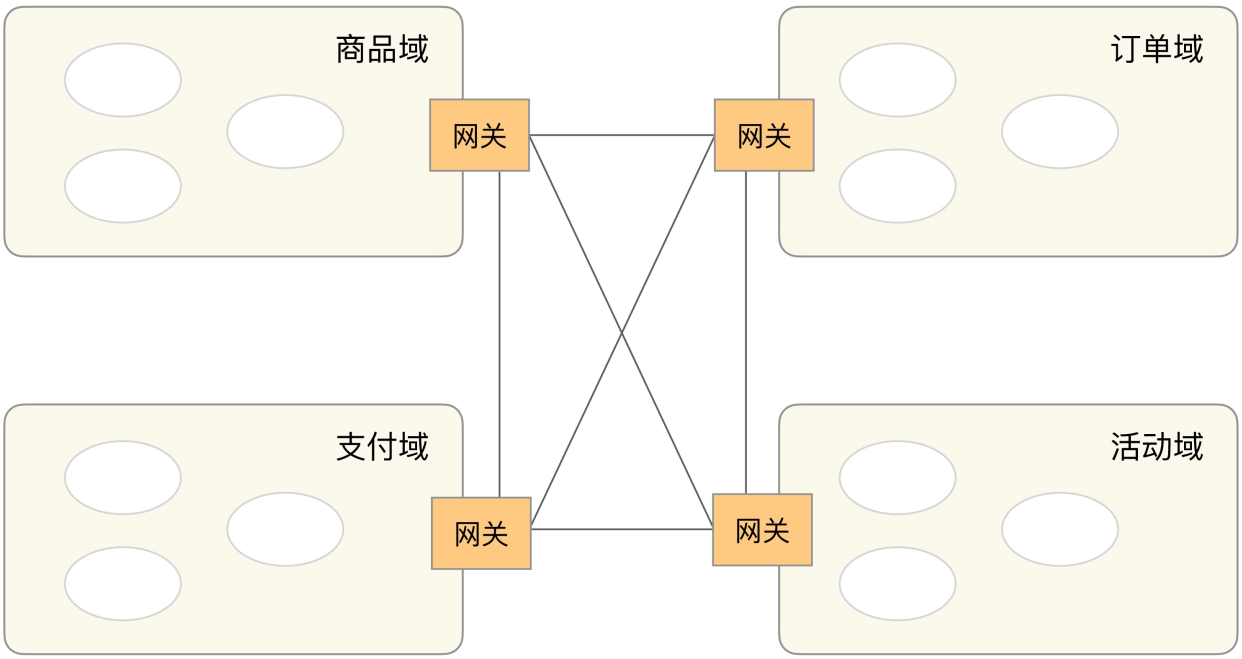
第二阶段：将商品和订单拆分到 2 个子系统里面。

第三阶段：商品子系统和订单子系统分别拆分成了更小的 6 个子系统。

上面只是个样例，实际上随着业务的发展，子系统会越来越多，据说淘宝内部大大小小的已经有成百上千的子系统了。

随着子系统数量越来越多，如果达到几百上千，另外一个复杂度问题又会凸显出来：子系统数量太多，已经没有人能够说清楚业务的调用流程了，出了问题排查也会特别复杂。此时应该怎么处理呢，总不可能又将子系统合成大系统吧？最终答案还是“合”，正所谓“合久必分、分久必合”，但合的方式不一样，此时采取的“合”的方式是按照“高内聚、低耦合”的原则，

将职责关联比较强的子系统合成一个**虚拟业务域**，然后通过网关对外统一呈现，类似于设计模式中的 Facade 模式。同样以电商为样例，采用虚拟业务域后，其架构如下：



小结

今天我为你讲了互联网架构模板中的用户层技术和业务层技术，希望对你有所帮助。

这就是今天的全部内容，留一道思考题给你吧，虚拟业务域划分的粒度需要粗一些还是要细一些？你建议虚拟业务域的数量大概是多少，理由是什么？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。（编辑乱入：精彩的留言有机会获得丰厚福利哦！）

精选留言 (38)

LouisLimTJ

2018-08-06

当然正确的话是要根据业务和团队来设计虚服务域。但是个人看法，粒度方面要粗一些，本来虚服务域就是来解决系统拆分过细的问题。
至于具体多少个为好，我们可以仿照管理学关于一个一层管理团队的理想大小，其答案不一定，但一般是不要超过10个，我个人比较舒服的数目是3到7个。

作者回复: 是的，5+-2的选择比较合理

共 2 条评论 >　　52

蚂蚁内推+v
2018-08-30

老师好 方便什么时候介绍下单点登录sso吗

共 1 条评论 >　　17

钱
2019-09-04

课后思考及问题
1：本文核心观点
1-1：面对庞大复杂的东西，用拆字决——化整为零，分而治之。
1-2：面对细微量多的东西，用合字决——分久必合，合二为一。
2：虚拟业务域划分的粒度需要粗一些还是要细一些？你建议虚拟业务域的数量大概是多少，理由是什么？
要粗一些，大概5～7个
理由：虚拟业务域要解决的是划分的系统太细太多的问题，关键是太细导致了太多，这是分而治之现在要合二为一，所以，最好粗一些，咱们的集体领导制最高领导人也就5或7个人，十几亿人民也是管理的好好的。划分的太少，容易集中，复杂化，划分的太多，容易分散，不好管理。

作者回复: 观点很独特啊😄

共 2 条评论 >　　9

lyonger
2020-03-22

一开始以为互联网架构模版中的技术，会介绍具体的技术知识，结果理论居多。😹

作者回复: 架构模板目的在于告诉读者一个公司的技术全貌，如果拆开来讲，每个都可以是一个专栏，你可以看看极客时间上其它专栏

共 2 条评论 >

7

小神david

2021-03-14

互联网业务千差万别万别，不过如果对比粗细来划分，虚拟域是要粗一些的，因为它出现的目标就是如此，整合更细粒度的微服务。具体数量这个没办法统一来分，结合实际业务情况和团队人员情况而定吧。

作者回复: 你可以从性能的角度来考虑，加入一个虚拟域处理需要50ms，10个就500ms了，因此我建议不要超过10个。

6

风雨无阻

2018-10-11

虚拟域是不是有点像中台了

作者回复: 不是中台，中台是不同业务的共性部分，虚拟域还是聚焦在本业务，当然，虚拟域可以依赖中台

5

l-m-a

2018-08-06

个人觉得增加了facade层，服务器机器数量提升，另外服务之间的调用并没有减少反而还增加了，facade层的性能直接影响内部服务的能力。

作者回复: 这也是代价，所以没有完美的解决方案
回到这个方案，当需要引入facade层的时候，服务器数量已经不是问题。另外，服务之间的调用不会减少还会增加，但是整个系统的关系复杂度会降低

5

Geek_b04b12

2018-08-08

有几个知识盲点：
1.facade模式，和工厂模式一个范畴？
2，虚拟业务域中图片中的网关，是啥意思？相对于域名来调用服务？还是别的？在阿里的接口和域名访问，用户能感受到吗？或者作为学习者有验证的端倪吗？

作者回复: 1. 设计模式里面有详细阐述
2. 这个网关是内部的，可以是protobuf这种协议，也可以是HTTP协议，用户不可见

3

Geek_steven_wang
2020-01-15

数量上应该根据具体业务领域逐层汇总，到达一个团队可接受的数量，一开始建议尽量少5个一下，要考虑后面业务扩展还会加，但最多不建议超过9。

另外问题就是一个域内的服务数量还是多，这个有什么办法吗？不能再汇总出网关了，这样层级太深。

作者回复: 答案就是域内网关，层级深但是对域外其它系统没什么影响

2

考休
2019-11-21

请问老师，目前大部分系统都是前后端分离的架构，前后台的调用使用jwt做token验证，在这种前提下，还需要考虑单点登录的方案吗，是不是所有的模块只需要采用相同的验证、生成规则就可以了，当然对于微服务架构的话，将鉴权放到网关上完成就可以了，感觉像cas这样的方法像是可以给之前jsp时代用的

作者回复: 现在微服务都要用单点登录哦

共 2 条评论 >

2

feifei
2018-08-10

这个肯定要粗一些，高内聚么，这样肯定功能类似的都被合并为一个了！我觉得控制在个位数吧！我觉得这是相对的一个平衡吧

作者回复: 是的，粗一些比较好，5+-2原则比较合适

2

张玮(大圣)
2018-08-08

既然上升到虚拟域了，粒度粗一些会好很多，细的话容易淹没在业务中，理不清域之间边界了。

从以前经历的一些项目得出的结论，10 个全是比较多了，再多的话就需要加入子域来说明了。

最后，一起陪伴走过这么多专栏课时，运华兄辛苦！

作者回复: 虚拟域粒度确实要粗一些。
感谢你的鼓励，再辛苦也值得😄

2

wuhulala
2018-08-07

我们现在比如说有一个a子系统里面有两个子工程b和c b和c单独提供服务 那么这样b和 c算是两个微服务 a算是虚拟业务域 如果是这样的话 我认为这样划分逻辑清晰 也无需多做什么工作有何不可呢。期待答疑

作者回复: 子工程不算微服务，微服务能独立部署和运行

2

谭方敏
2020-03-14

用户层技术
用户管理
单点登录sso，又叫统一登录。
单点登录的技术实现包括cookie，jsonp，token等，比较成熟的开源单点登录方案当属cas。
在业务变大，成长为平台后，又引出了用户管理的第二个目标：授权登录。现在流行的就是oauth2.0协议。

消息推送

消息推送根据不同途径，分为短信，邮件，站内信，app推送。除了app推送以外，不同的途径基本上调用不同的app即可完成。

重点的app推送包括ios推送，以及android推送，ios一般用apns推送就好了，而android就不一样了，在国外google cloud message是可以用的，但是在国内是没法用的。所以就衍生了各种版本，主要有大厂自己实现的消息推送机制（阿里云移动推送，腾讯信鸽推送，百度云推送），以及专业的第三方公司提供的推送服务（友盟推送，极光推送）。

消息推送主要包括三个功能：设备管理（唯一标识，注册，注销），连接管理和消息管理器

如果要自己处理的话，那么要注意三点挑战。

海量设备和用户管理

为方便业务推广，还需要收集用户信息，将用户跟设备关联，提取用户特征对用户进行分类和打标签。

连接保活

一般设备为了节省电（流）量，都会限制应用后台运行，限制应用后台运行连接通道可能被中断，导致消息不可达。方法有很多，比如互相拉起，找手机厂商开白名单。

消息管理

实际业务中，需要根据用户的特征选择一些用户进行消息推送，由于用户特征变化大，需要将推送给哪部分用户的逻辑设计非常灵活，才能满足需求。

存储云，图片云

互联网场景下，用户会上传多种类型的文件数据，这样文件的特点：

1）数据量大，用户基数大

2）文件体积小，几百kb到几mb，

3）访问有时效性，大部分文件是刚上传时访问最多，随着时间的推移访问量越来越小。

存储云/图片云都是基于cdn+小文件存储的技术，没有合并的原因在于图片业务复杂还包括裁剪，压缩，美化，审核，水印等处理。

业务层技术

随着业务复杂度越来越高，一般的方法就是拆，化整为零，分而治之，将整体复杂性分散到多个子业务/子系统里面去。

当然随着子系统数量越来越多，还可以采用合的做法，比如，按照高内聚，低耦合的原则。将职责关联比较强的子系统合成一个虚拟业务域，通过网关对外统一呈现，类似于facade设计模式。

虚拟业务域划分应该越细越好，因为要按照高内聚低耦合的原则，虚拟业务域数量一般3-5个？具体原因不太清楚，平常业务中没有涉及到。。。。

1

客舟听雨来coding
2019-10-05

这虚拟域感觉有些像领域驱动设计的限界上下文

作者回复: 大道相通😄

1

Sam.张朝
2019-09-29

原来最初接触微服务，只知道要拆分，不能拆分过细。
现在是要按照业务域拆分合并。

作者回复: 很多公司都经历过拆然后合并

共 2 条评论 >

1

wikili
2019-01-24

在同一个虚拟业务域下合并的两个业务子系统自我调用怎么实现的？比如A子系统需要调用B子系统

作者回复: 微服务调用，需要服务注册发现

1

蚂蚁内推+v
2018-08-30

老师好 方便介绍下单点登录吗

作者回复: 你需要什么程度的介绍呢？专栏已经简单介绍了😄

1

文竹
2018-08-26

首先根据团队人数来决定业务域的数量，一个业务域可以由3个开发人员负责。如果有12个开发人员，那么就划分成4个业务域，再来看是用粗还是细粒度划分。

作者回复: 12个人还远远不到划分业务域的时候，划分微服务就可以了

1

张玮(大圣)

2018-08-08

既然上升到虚拟域了，粒度粗一些会好很多，细的话容易淹没在业务中，理不清域之间边界了。

从以前经历的一些项目得出的结论，10 个全是比较多了，再多的话就需要加入子域来说明了。

最后，一起陪伴走过这么多专栏课时，运华兄辛苦！

1