

电商系统演进

学习目标:

- 了解电商业务的发展历程
- 了解电商相关的业务模式
- 学习电商中台系统的相关内容
- 掌握电商网站的架构演进

1. 业务模式演进

1.1 发展历程

1.1.1 萌芽期 (96-99)

- 96年: 国家信息化领导小组成立
- 97年4月: 各省成立信息化小组
- 97年12月: 中国化工网B2B上线
- 98年3月: 第一笔互联网交易完成 (个人第一笔交易经历?)
- 98年11月: 腾讯成立
- 99年5月: 8848网成立
- 99年8月: 易趣网
- 99年9月: 阿里巴巴
- 99年11月: 当当网

1.1.2 发展期 (00-10)

- 00年4月: 慧聪网B2B
- 00年5月: 卓越网 (现: 亚马逊中国) 开启B2C模式
- 00年6月: 中国电子商务协会成立
- 01年: 13所高校开启电子商务专业
- 01年11月: 8848暂停电子商务
- 02年3月: eBay收购易趣33%的股份
- 02年10月: 阿里实现收支平衡
- 03年5月: 淘宝网成立, C2C
- 03年6月: eBay全盘收购易趣C2C
- 03年10月: 阿里推出支付宝
- 03年12月: 慧聪网香港上市
- 04年1月: 京东涉足电子商务
- 04年8月: 亚马逊收购卓越
- 05年9月: 腾讯推出拍拍网 (拍拍网的使用经历?)

- 06年5月: 淘宝推出淘宝商城
- 07年11月: 阿里巴巴登陆香港证券市场
- 08年: 经济危机部分严重依赖外贸的电商倒闭
- 09年6月: 当当宣布首季盈利

- 10年1月：苏宁易购上线
- 10年11月：国美电器控股库巴购物网进军电子商务

1.1.3 稳定期（11-今）

- 12年1月：淘宝商城改为天猫
- 12年3月：唯品会上市
- 19年：天猫双11交易额2684亿
- 11年至今：天猫、京东、苏宁、国美、各大电商趋于稳定

1.2 业务模式

电商早期多以单体业务为主，逐个业务线扩张。系统也多呈现为多个mvc独立运行状态。下面逐个介绍各个单体的业务模式，以及他们各自的系统运行特点。

1.2.1 B2C

1) 简介

Business to Consumer (Customer)，B2C中的B是Business，意思是企业，2则是to的谐音，C是Customer，意思是消费者，所以B2C是企业对消费者的电子商务模式。这种形式的电子商务一般以网络零售业为主，主要借助于Internet开展在线销售活动。

2) 系统特点

因为面向大量消费者，网站访问量较大，对网站并发行有一定要求，但交易方式相对简单。

3) 案例

天猫商城

京东

苏宁易购

国美电器

1.2.2 C2C

1) 简介

Consumer to Consumer，意思是个人与个人之间的电子商务。C2C 商务平台就是通过为买卖双方提供一个在线交易平台，使卖方可以主动提供商品上网拍卖，而买方可以自行选择商品进行竞价。比如一个消费者有一台电脑，通过网络进行交易，把它出售给另外一个消费者。

2) 系统特点

同B2C一样，网站访问量是重点。

3) 案例

闲鱼

58交易

1.2.3 B2B

1) 简介

Business to Business, 即企业与企业之间通过互联网进行产品、服务及信息的交换。通俗的说法是指进行电子商务交易的供需双方都是商家（或企业、公司）。一般来讲，过程包括：发布供求信息（现货，期货），订货及确认订货（议价，集采，聚投），支付（先货后款，先款后货，分期支付），票据的签发、传送和接收（验货，验票），确定配送方案（物流），监控配送过程（违约与仲裁）等。

2) 系统特点

与C端用户不同，B端访问量相对小，但是交易周期的流程，以及交易的监督相对复杂。

3) 案例

中国化工

阿里巴巴

慧聪网

中国供销电子商务

中粮我买网

1.2.4 O2O

1) 简介

Online to Offline, O2O 是新兴起的一种电子商务新商业模式，即将线下商务的机会与互联网结合在了一起，让互联网成为线下交易的工具。这样线下服务就可以用线上来引流，消费者可以用线上来筛选服务，还有成交可以在线结算，高效完成部分线下的前期环节。

2) 系统特点

O2O整合了线上购买+线下体验的模式，多为服务性商品，比如餐饮，娱乐等。尤其是移动互联网的接入，对O2O系统就有针对用户做定制化服务的要求，比如位置定位、周边服务、定向推送等。

3) 案例

各类团购

外卖网

打车类

共享单车

1.2.5 其他

1) ABC

Agent、Business、Consumer。ABC 模式是新型电子商务模式的一种，被誉为继阿里巴巴 B2B 模式、京东商城 B2C 模式以及淘宝 C2C 模式之后电子商务界的第四大模式。它由代理商、商家和消费者共同搭建的集生产、经营、消费为一体的电子商务平台。

2) B2M

Business to Manager。B2M 相对于 B2B、B2C、C2C 有着本质的不同，其根本的区别在于目标客户群的性质不同，前三者的目标客户群都是作为一种消费者的身份出现，而 B2M 所针对的客户群是企业或者该产品的销售者或者为其工作者，而不是最终消费者。

3) M2C

Manager to Consumer。M2C 是针对 B2M 的电子商务模式而出现的延伸概念。B2M 环节中，企业通过网络平台发布该企业的产品或者服务，职业经理人通过网络获取该企业的产品或者服务信息，并且为企业提供产品销售或者提供企业服务，企业通过经理人的服务达到销售产品或者获得服务的目的。

4) B2A/B2G

A: Administration, G: Government。是企业与政府管理部门之间的电子商务，如政府采购，海关报税的平台，国税局和地税局报税的平台等。

5) C2A/C2G

Consumer to Administration。消费者对行政机构的电子商务，指的是政府对个人的电子商务活动。例如政府的税务机构通过指定私营税务，或财务会计事务所用电子方式来为个人报税。

1.3 电商中台

1.3.1 背景

(供销电商的真实经历，背景，发展与问题?)

1) 技术架构的需要

烟囱式系统建设：业务部门提出业务需求，信息中心部门进行系统集成，再进入到需求收集、需求分析、开发、测试、上线的项目周期中。每个新系统的上线都预示着一座新的烟囱矗立而成。这种完全基于业务需求建设系统的方式，已经成为过去20多年企业建设IT系统的标准流程，导致IT系统建设早的企业内部系统烟囱林立。

造成的问题：

- 重复功能建设和维护带来的重复投资
- 系统间的集成和协作成本高昂

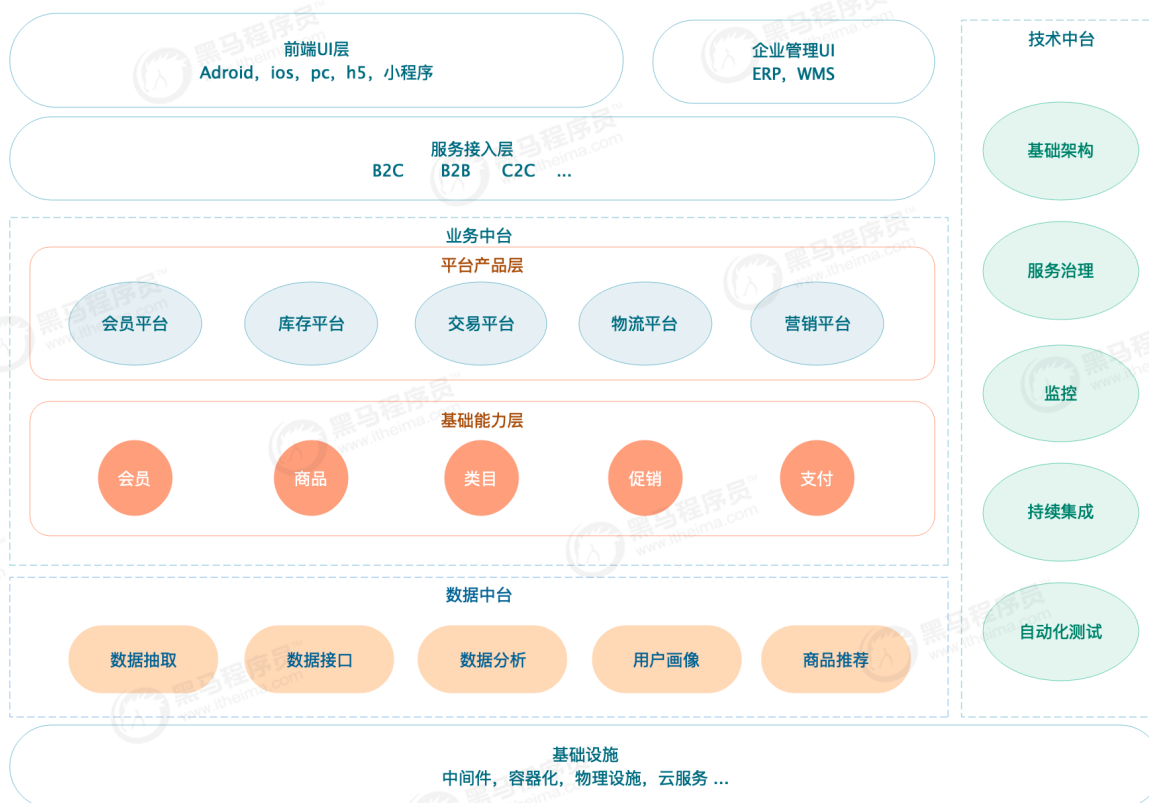
- 不利于业务的沉淀和持续发展。

2) 组织架构的需要

IT信息部门在现有模式下往往被定位成了一个服务部门。这使得IT信息化部门一直处于『业务支持』的职能位置，即只为了满足业务部门需求而进行IT系统建设的实施和运维部门。这种模式下，很多企业的IT信息化部门的员工大多数的内容都是开发上线开发上线。而对业务缺乏某一专业领域的经验和沉淀。基于中台下的共享业务技术部，使得核心公共业务沉淀，让IT部门与上层业务更贴近，能够对业务的下一步发展有着自己的理解和看法，对业务流程如何进一步优化能更好地提升业务，甚至对企业现有的业务提出创新的想法，为企业带来新的业务增长点。

1.3.2 概述

综合前面的业务模式，各个业务类型中，都具备基本的商品、交易、库存、支付等公共部分。提炼这部分基础内容，进行沉淀，逐步形成中台基础能力层，而个性化的业务流程部分上浮，形成产品层。这样的好处是，基础能力层聚焦于稳定收敛的业务模型和基础服务本身，不会随着业务和前台产品的调整发生大的变化，平台产品层则专注于通过流程编排类的技术手段，将基础能力构建成为业务的解决方案，解决共性和个性化的问题。即大中台，小前端。



1.3.3 业务中台

- 商品中心：商品、类目、sku、spu
- 交易中心：订单、状态流转、条目、支付
- 营销中心：促销、优惠券、活动
- 会员中心：账户、基本信息、收发货地址、商铺商家信息
- 仓储中心：仓库、库存
- 物流中心：发货信息、自主物流或外部物流对接

1.3.4 技术中台

- 基础架构：核心类库、公共框架、基础服务、服务治理框架
- 中间件：分布式缓存、分布式消息、数据存储（db, nosql）、分布式文件、分布式调度
- 自动化运维：监控中心、资源管理、配置中心、发布中心、日志平台
- 自动化测试：任务协同、基础测试、性能测试、接口测试、持续集成

(运维中台的争议?)

...

1.3.5 数据中台

- 数据抽取：从db, nosql, 日志等各个来源提供抽取接口
- 数据接口：为上层业务提供需要的定制化业务数据接口
- 数据分析：行业分析与决策、数据驱动运营
- 人工智能：用户画像、商品推荐
- 可视化：数据大屏、信息展示、活动报表

...

1.4 发展趋势

1.4.1 移动电商

移动电子商务（M-Commerce），它由电子商务（E-Commerce）的概念衍生出来，传统电子商务以PC机为主要界面，是“有线的电子商务”；而移动电子商务，则是通过智能手机、平板电脑这些可以装在口袋里的终端与我们谋面，无论何时、何地都可以开始。有人预言，移动商务将决定21世纪新企业的风貌，也将改变生活与旧商业的地形地貌。

主要特点：随时随地、实时在线、精准推送营销

面临挑战：终端的多样化

1.4.2 社交电商

电商说到底就是通过线上流量把货卖出去。社交电商顾名思义，与传统电商网站的区别在于流量的导入方式是借助社交圈。社交电商，一般通过点赞、分享、转发、自媒体推荐等通过自己的社会关系去制造和产生流量。而传统电商是做平台，借助平台的力量去助推销售。

传统电商以货为中心，围绕商品、供应链的传统卖货平台。

社交电商以人为中心，是社交关系形成的电商形态，不以产品搜索、展示为销售模式，而是通过社交，用户分享传播，形成口碑效应，从而激发消费需求。

拼多多历经短短2年3个月的时间就在美国纳斯达克正式上市，而这个成绩，淘宝用了10年，京东用了5年。

案例：拼多多、有赞、微商、抖音、淘宝直播

特点：容易形成圈子效应

面临挑战：对口碑的要求很高、社交和交易的界限不好把控

1.4.3 新零售

新零售，英文是New Retailing，在2016年的10月，马云提出了新零售就是线上、线下、物流相结合。即个人、企业以互联网为依托，通过运用大数据、人工智能等先进技术手段，对商品的生产、流通与销售过程进行升级改造，进而重塑业态结构与生态圈，并对线上服务、线下体验以及现代物流进行深度融合的零售新模式。

在线上零售的冲击下，传统的纯线下零售在历经了地产整合、渠道升级、品牌崛起等发展阶段后，很难有新突破。而线上零售也逐渐探到传统流量模式的天花板。于是线上、线下、技术、数据、供应链等场景都在寻求相互融合，形成线上交易结合线下体验店的新零售。

案例：阿里巴巴全面布局"新零售"市场；京东主推"无界零售"；苏宁则依靠"智慧零售" 迎来九年来最佳业绩

2. 架构体系演进

2.1 概述

任何体系的成型不是一蹴而就，随着访问量，数据量的增长，业务需求在推动技术架构的发展变革。下面我们以淘宝的发展历程为例，来看系统架构的演进过程。

1) 架构目标

高性能：提供快速的访问体验，高并发下的及时响应。

高可用：网站服务7x24正常访问，可用性达到几个9。

可伸缩：资源的扩容，应对突发和流量脉冲。

安全性：提供网站安全访问和数据加密、安全存储等策略。

扩展性：方便对现有模块做版本升级，新模块的上线，突发活动下的服务降级。

敏捷性：对系统突发情况的快速排查与应对。

2) 演进概述

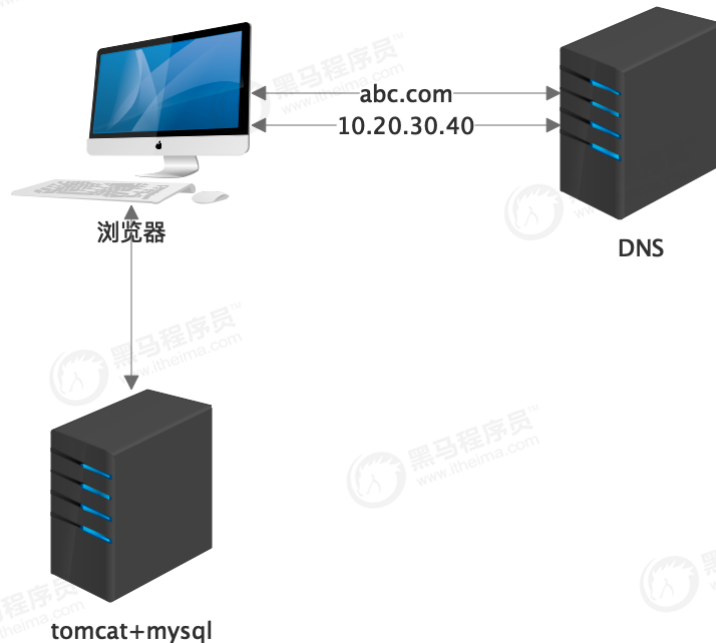
部署层面：单机到集群，集中式到分布式，物理部署到云化

业务层面：单一mvc到垂直拆分，服务治理到微服务

数据层面：db到集群，单一关系型数据到多样化nosql，搜索引擎，文件服务

2.2 单机器时代

小型网站，阿里云小项目还有人在用。



1) 方案

大型机：引发对单机性能的过度追求，推动高配机器的发展，成本高昂

调优：jvm单节点调优甚至接近于强迫症的地步

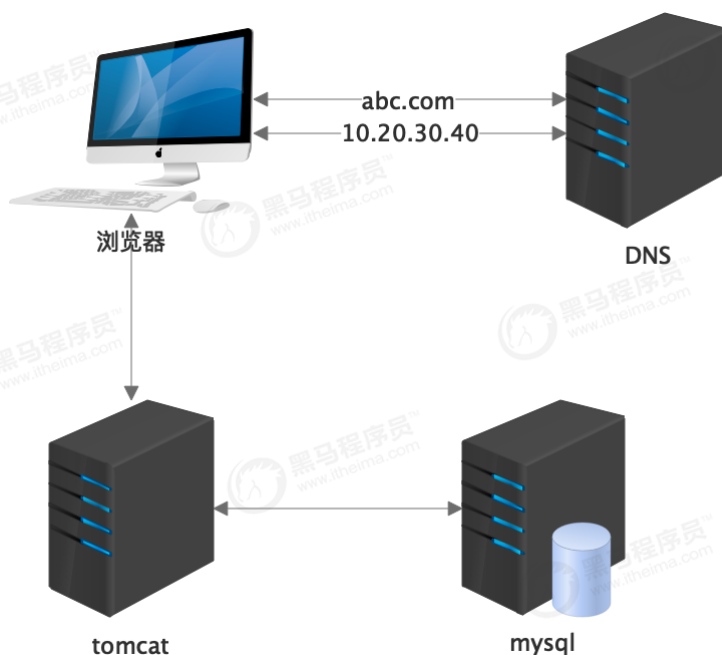
2) 特点

部署简单：采用web包部署与发布，db同台机器连接，简单易操作。

资源争夺：在业务发展的初始阶段尚可支撑，随着访问量的上升，单机性能很快会成为系统瓶颈。

2.3 数据分离

稍微大一点的系统，dba出现，数据库追逐商业大型db如oracle，如（淘宝v1.1，mysql→oracle）



1) 方案

多台机器：tomcat与mysql各自独占机器资源

针对性扩容：tomcat应用机更注重cpu的运算和内存，mysql更注重io与磁盘性能，针对各自情况扩容

2) 特点

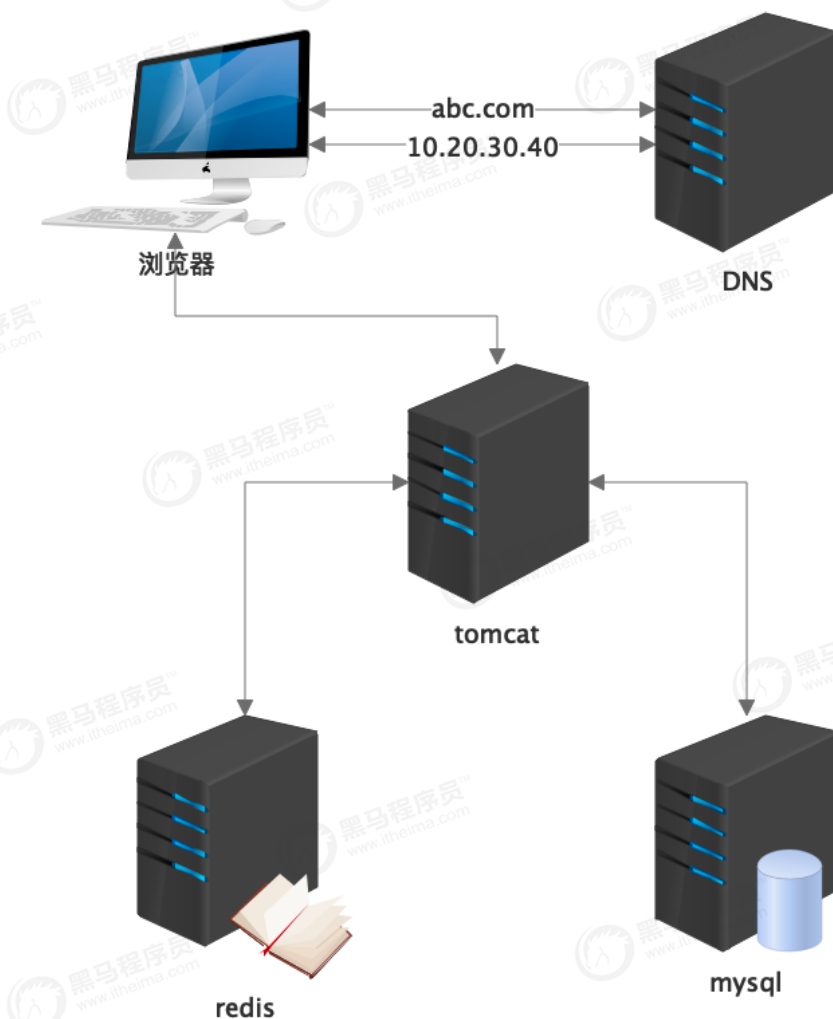
数据维护：可以抽出单独的dba来维护数据库服务器

数据安全：需要跨机器访问数据库，链接密码需要注意防范泄漏

数据库瓶颈：数据库频繁读写，io很快成为瓶颈

2.4 数据缓存

2006-2007，淘宝V2.2架构，分布式缓存Tair引入。（08-09创业初期memcache+ssh1时代的故事）



1) 方案

数据冷热划分：热点数据如类目、商品基础信息热加载，其他数据延迟加载

ehcache：非分布式，简单，易维护，可用性一般

memcache：性能可靠，纯内存，客户端需要自己实现，无持久化

redis: 性能可靠, 纯内存, 自带分片, 集群, 哨兵, 支持持久化, 几乎成为当前的标准方案

2) 特点

缓存策略: 缓存与db的边界需要架构师去把控, 重度依赖可能引发问题

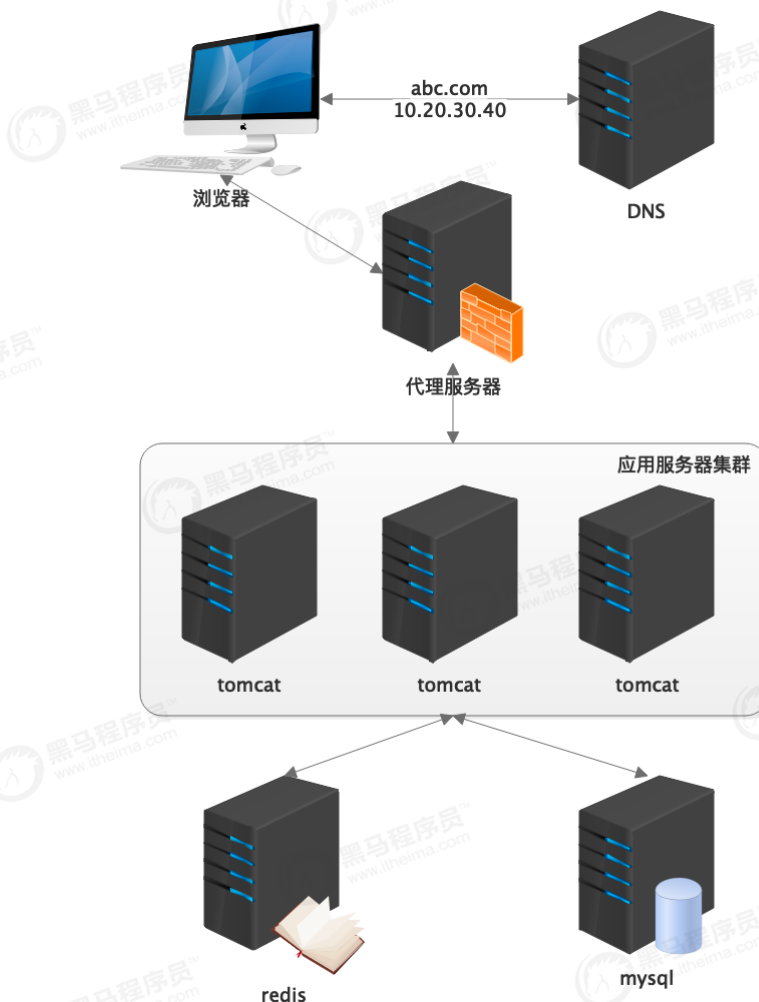
(memcache造成db高压案例, 模拟请求解决 → squid; redis短信平台故障案例)

缓存陷阱: 击穿, 穿透, 雪崩

数据一致性: 删除、双写

2.5 应用集群

2004-2005, 淘宝V2.0, EJB为核心 (2011年间EJB3 pk spring3.x选型案例)。V2.1架构下, 引入spring框架走向轻量化和集群



1) 方案

apache: 早期负载均衡方案, 性能一般

nginx: 7层代理, 性能强悍, 配置简洁, 可以携带lua完成前端逻辑, 当前不二之选

haproxy: 性能没问题, 可做7层或4层代理。

lvs: 4层代理, 性能最强, linux集成, 配置麻烦

h5: 硬件负载，财大气粗的不二选择

2) 特点

session保持: 集群环境下，用户登陆及session的保持成为问题，需要分布式session做支撑

分布式协同: 分布式环境下对资源的加锁要超出线程锁的范畴，上升为分布式锁

调度问题: 调度程序跑重复

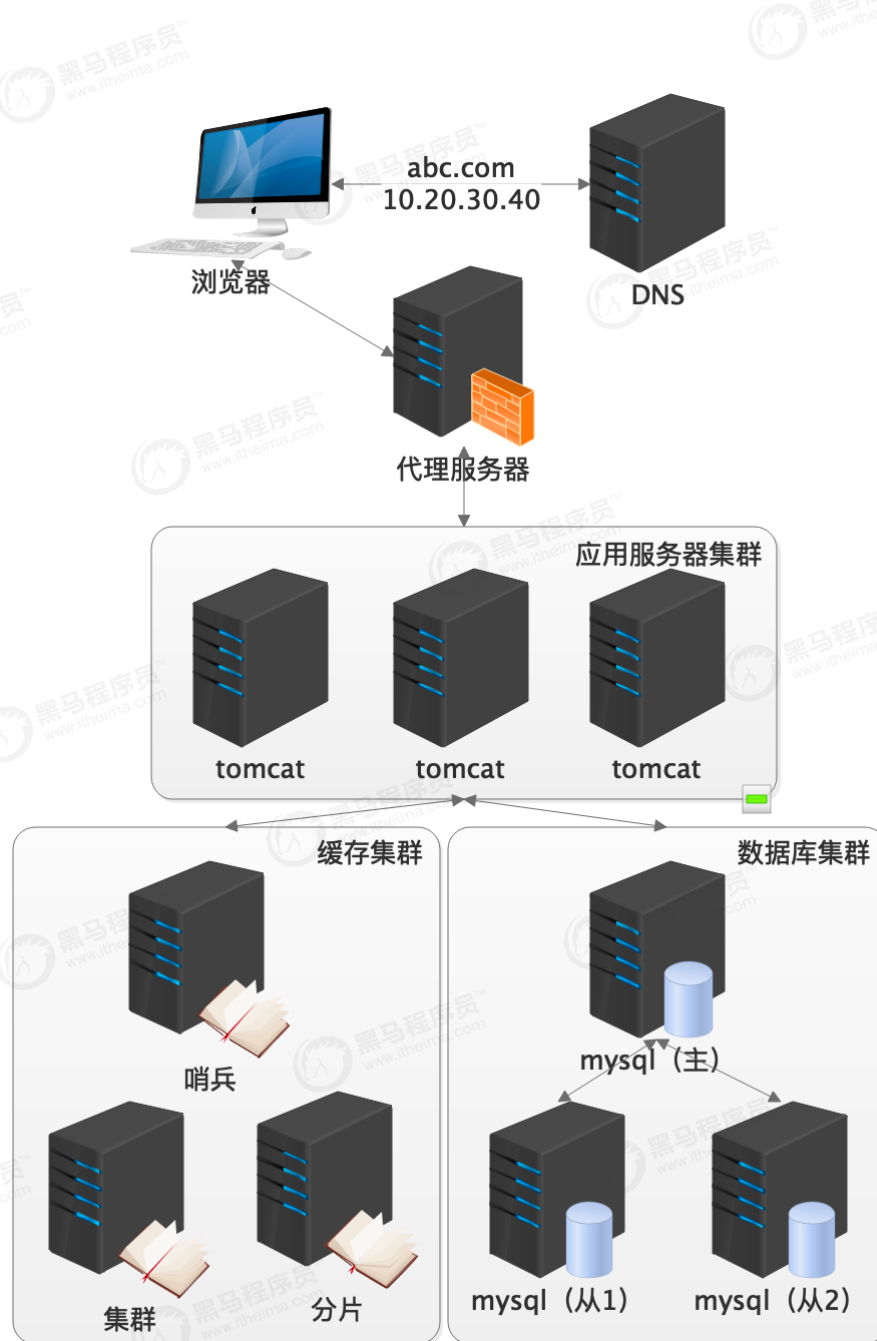
机器状态管理: 多台应用机的状态检测与替换需要做到及时性

服务升级: 滚动升级成为可能

日志管理: 日志文件分散在各个机器，促进集中式日志平台的产生

2.6 读写分离

早在2003-2004淘宝V1.0就使用mysql就使用了读写分离，V1.1换成oracle，直到2007数据库重新往mysql回迁，新东方也是相同经历。



1) 方案

缓存集群：redis哨兵，集群，分片，pre-sharding，memcache一致性hash

数据库集群：一主多从、双主单写、灾备（供销灾备双主单写案例）

2) 特点

数据延迟：准实时，单方法内的写入立即读取问题

开发层面：需要开发框架具备多数据源的支持，以及自动化的数据源切换

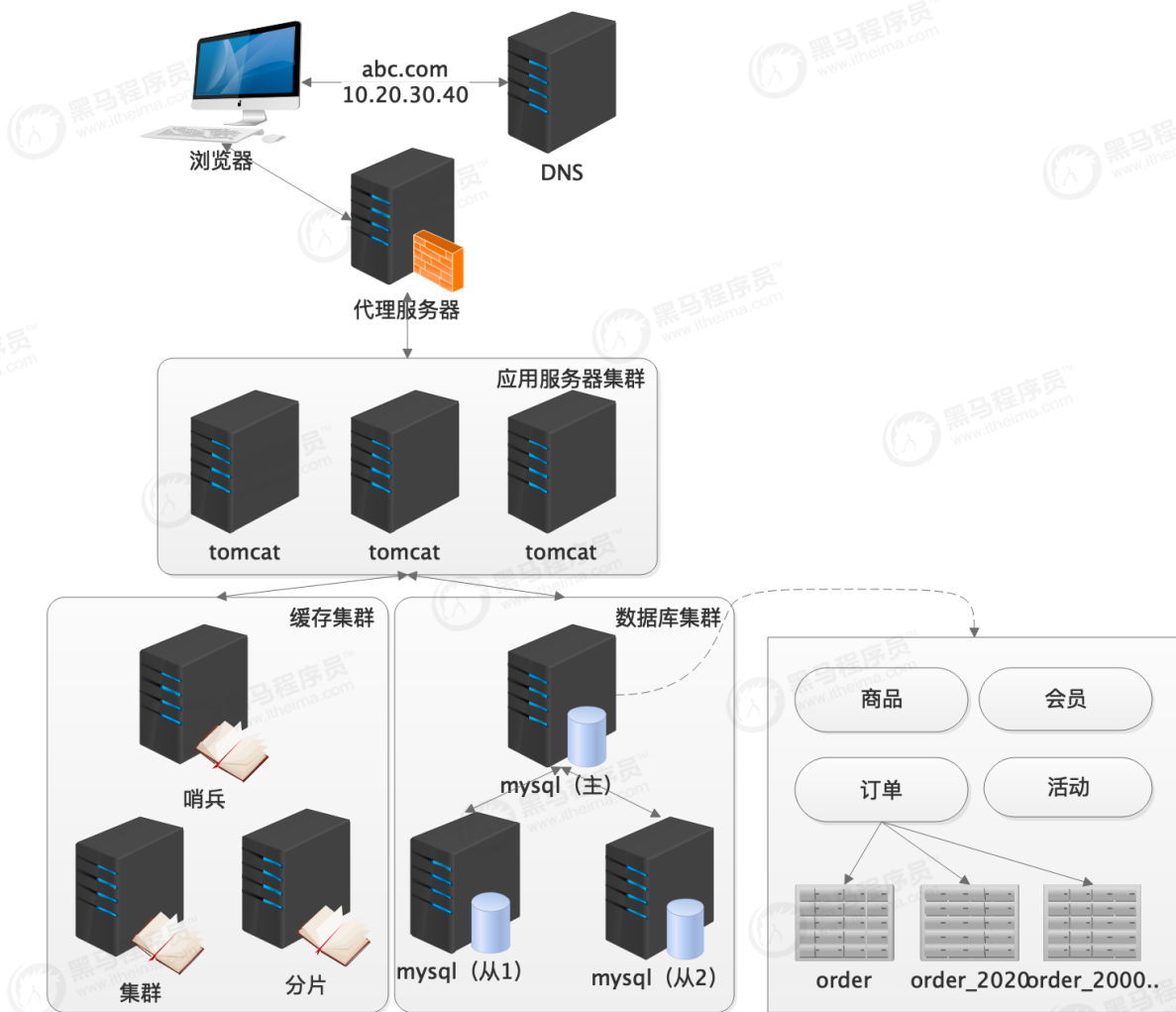
数据分片：memcache需要客户端处理，redis支持集群数据分片

单库瓶颈：业务越来越多，表数量越来越多。单库成为瓶颈

数据局限：依然无法解决单表大数据的问题，比如订单积累达到亿级，即使在从库，关联查询依然奇慢无比

2.7 分库分表

2004-2007, 淘宝V2.1, 支持分库, 抛弃EJB。



1) 方案

早期分区表: range, list, hash, key, 对开发端透明, 但分区数有限制, 性能提升有天花板。

业务分库: 订单库, 产品库, 活动库, 会员库

横向分表: 3个月内订单, 半年内订单, 更多订单

2) 特点

(恒天集团基金系统从数据库分区表到Mycat)

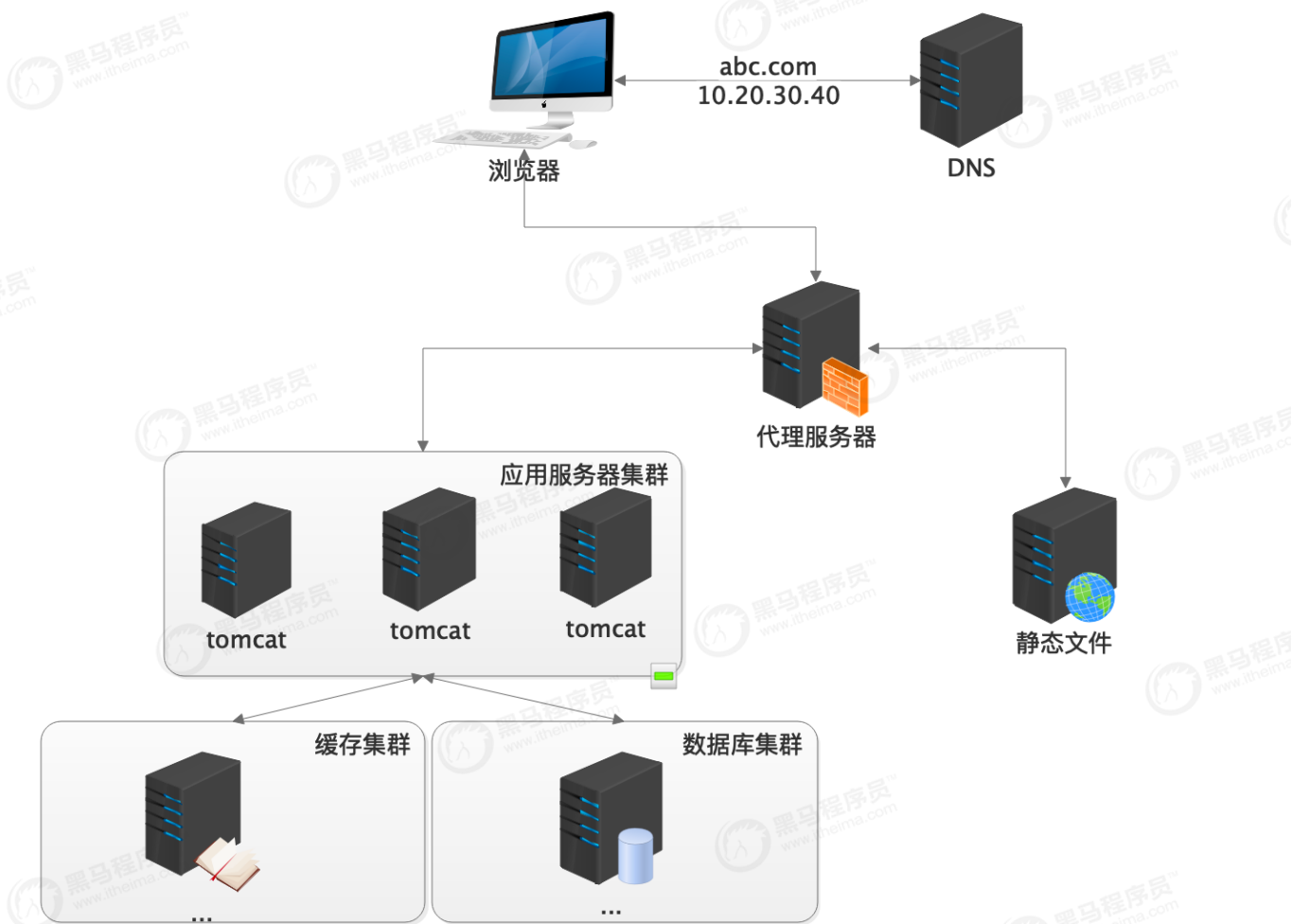
分库: 无法使用数据库事务保证完整性, 而分布式事务的效果并不理想, 多采用幂等和最终一致性方案。

分表: 数据聚合矛盾, 以订单为例, 下单时间维度的分表和用户维度的查询是一对矛盾。排序统计变得异常困难。

中间件: Sharding-JDBC, Mycat, Atlas

2.8 动静分离

早年的Apache+tomcat，后被nginx几乎一统江山。（前后端开发模式的演进：mvc页面嵌套→接口化）



1) 方案

静态响应：tomcat对静态文件响应一般，提取静态文件，直接由nginx响应

动态代理：后端api通过代理转发给tomcat应用机器

2) 特点

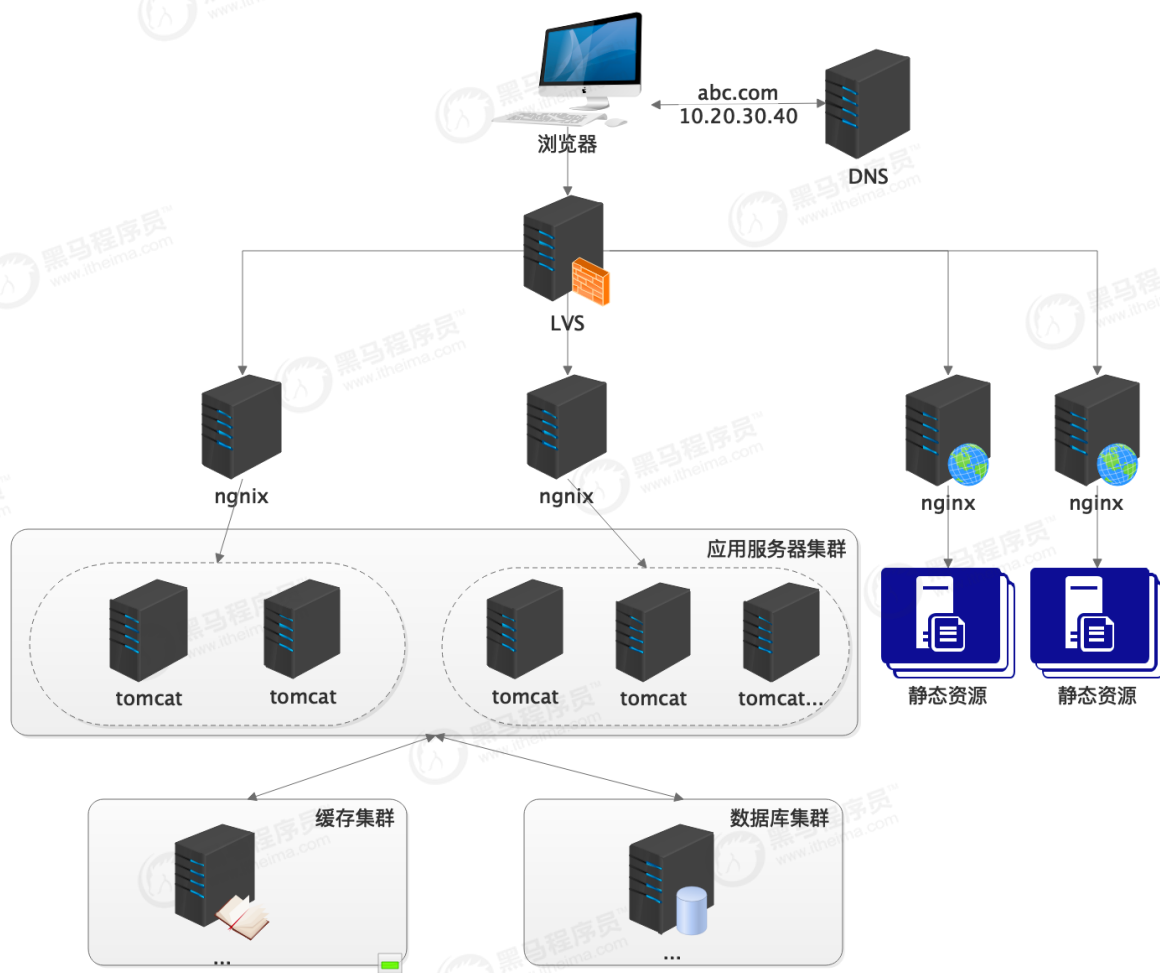
开发层面调整：项目结构要同步调整，由原来的一体化mvc转换为后端api+前端形式。

前后协调：前后端的分工变得更明确，互相并行开发，独立部署，但也带来了接口协调与约定等沟通问题

单层局限：单个nginx代理在并发处理任务时，依然会有上限，静态文件节点需要面临扩容。

2.9 多层代理

2010-2012，新东方网络课堂项目架构，基于springMVC+Mybatis，war包集中式部署。资源不够，机器来凑的时代（30台tomcat）。



1) 方案

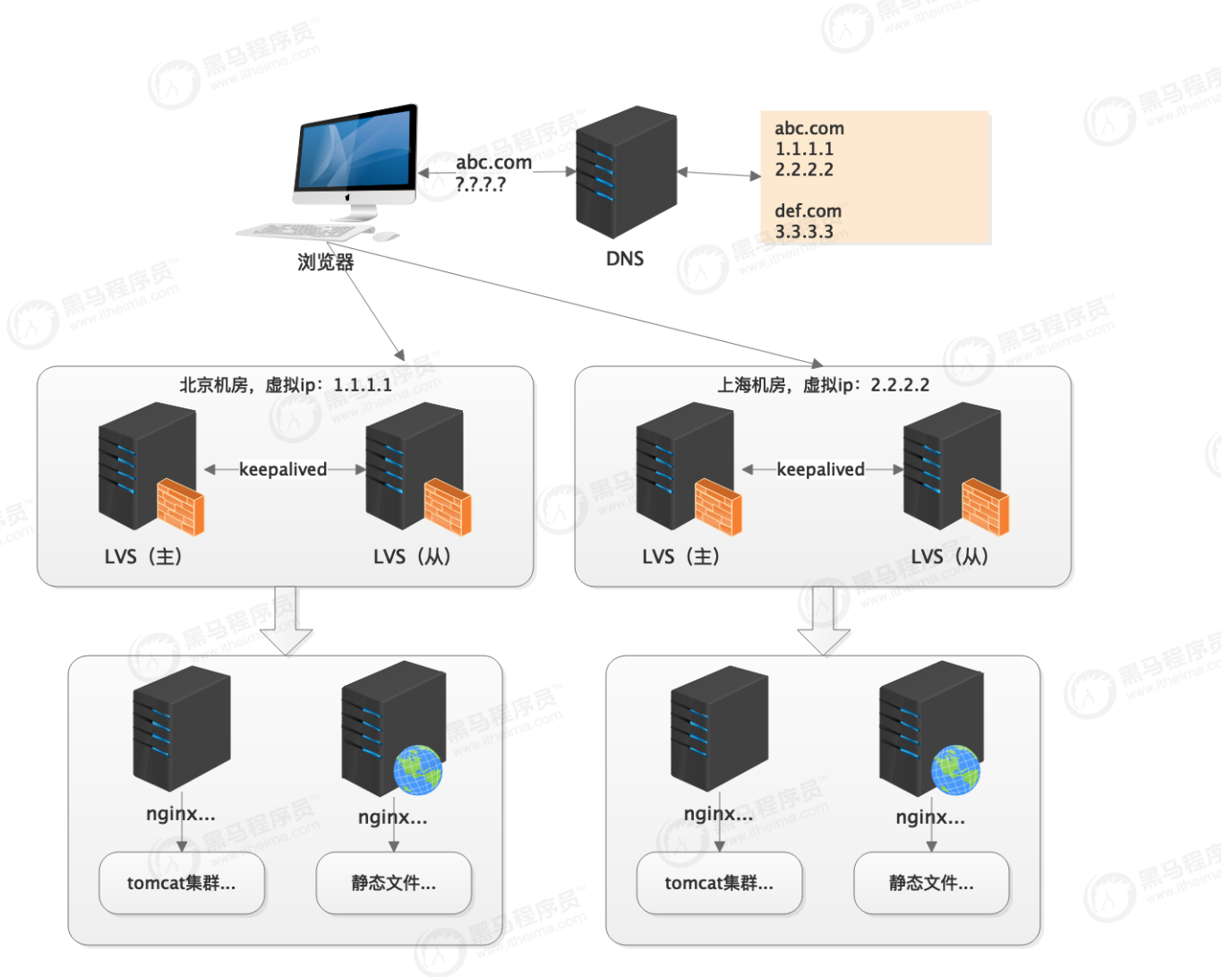
多层代理：在nginx前再加一层lvs做代理，作为流量的统一入口，再分发给下层的多个nginx，静态服务得到扩容

2) 特点

机房受限：lvs依然是单一节点，即使keepalived做到高可用，流量仍然需要在唯一入口进入。

2.10 跨机房

淘宝V2.1时代，使用自己的TaobaoCDN。中国供销集团两地灾备，DNS轮询北京机房，西安机房



1) 方案

dns轮询：通过配置多个ip将服务部署到多个机房，通过dns的策略轮询调用，可以实现机房层面的扩容

CDN：就近原则，使用户获得就近的机房访问相关资源，自己投资太大，购买他方需要付费。

2) 特点

基本解决了机器部署的扩容问题，随着业务的发展，数据呈现多样化发展，底层异构化数据成为新的瓶颈。

2.11 异构数据

2006-2007，淘宝V2.2，分布式存储TFS，分布式缓存Tair，V3.0加入nosql Cassandra，搜索引擎升级

数据库查询调优极限→搜索引擎、本地上传+nfs→文件系统的演进，方案后期均有深入讲解



1) 方案

nosql: 商品特殊化属性, mongodb, hbase

搜索引擎: 商品信息库, lucene, solr, elasticsearch

分布式文件存储: 商品图片, 上传的文件等, hdfs, fastdfs

2) 特点

开发框架支持: 存储的数据多样化, 要求开发框架架构层面要提供多样化的支撑, 并确保访问易用性

数据运维: 多种数据服务器对运维的要求提升, 机器的数据维护与灾备工作量加大

数据安全: 多种数据存储的权限, 授权与访问隔离需要注意

2.12 业务线拆分

以上架构的演进, 基础设施层面的优化几乎达到了天花板, 接下来, 需要从业务和应用层面进行架构上的升级



1) 方案

业务分发：代理层设置不同的二级域名，如b2b.abc.com，b2c.abc.com，分发给不同的服务器

消息互通：服务之间使用mq等异步消息提供通讯。

跨域问题：因为多个业务线占据不同的域名，出现多个主站，单点登录被推上前线

2) 特点

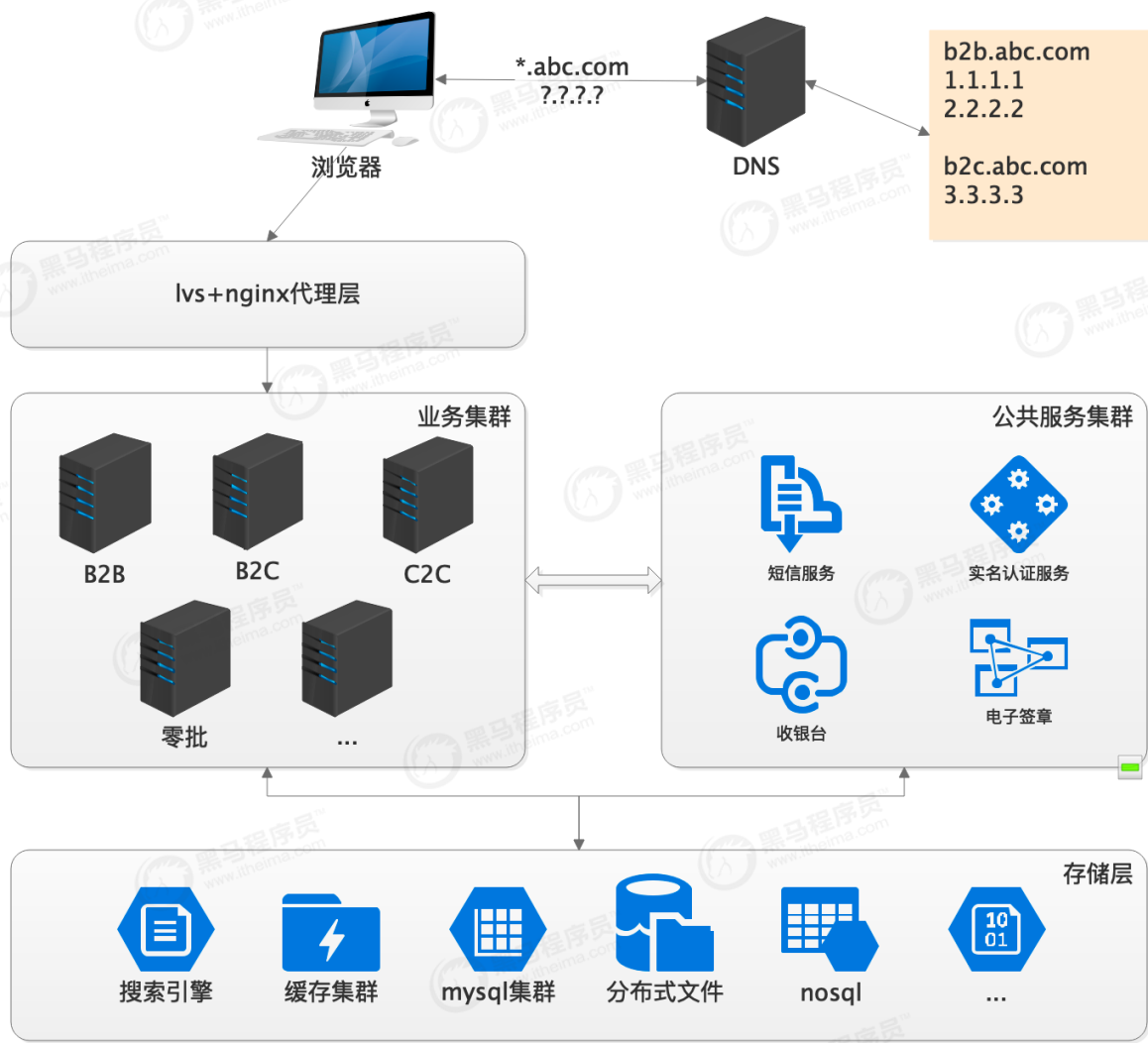
粒度较粗：纯以业务为导向，往往形成业务团队各自为战，新业务线出现时疯狂扩张

重复开发：相同功能可能在不同业务的项目中被重复开发，比如促销、短信发送、收银台

2.13 服务化

淘宝V3.0，HSF出现，服务化导向，架构师忙于SOA和系统关系的梳理。

(2015年冬金融项目业务线rest→dubbo2.4.11的引入过程)



1) 方案

公共服务：重复开发的基础服务沉淀，形成服务中心，避免重复造轮子，降低成本，架构团队出现。

独立性：各自服务独立部署升级，粒度更细，低耦合，高内聚

SOA：服务治理的范畴，重在服务之间的拆分与统一接口

微服务：可以理解为服务治理的一种手段，趋向于小服务的独立运作与部署。

技术手段：springcloud, dubbo

2) 特点

界限把控：服务的粒度、拆分和公共服务提炼需要架构师的全局把控。设计不好容易引发混乱

部署升级：服务数量增多，自动化部署面临挑战

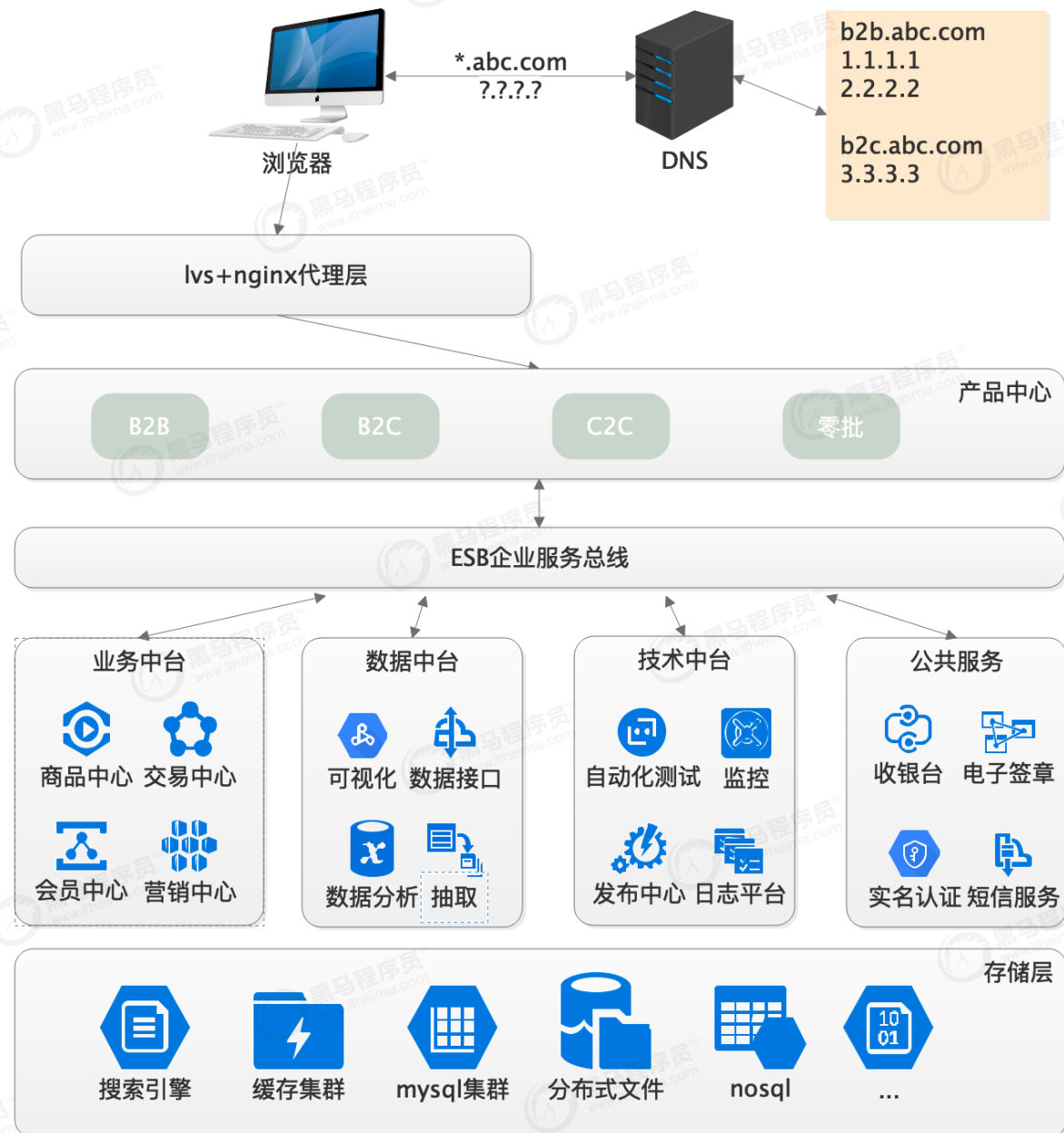
服务可用性：抽调的微服务因需要被多个上层业务共享，可用性等级变高，一旦down机就是灾难

熔断和限流：做好服务熔断和限流，提防服务单点瓶颈造成整个系统瘫痪。

2.14 中台化

阿里共享业务技术部的发展，中国供销集团，电商平台中台体系的架构模式。

技术沉积形成了公共服务平台，业务沉积逐步形成共享业务技术部，同时，业务烟囱的壁垒推动业务中台成型。同时组织结构同步升级，以技术共享为核心的技术中台，以数据为中心的数据中台同步建设得到实施。



1) 方案

业务沉淀形成独立的中心，各个中心之间借助服务总线实现业务协作与服务重组。

2) 特点

团队规模：团队规模扩张，人员增多，协调成本上升，组织机构要有配套调整

接口授权：各个中心的接口授权与开发需要把控

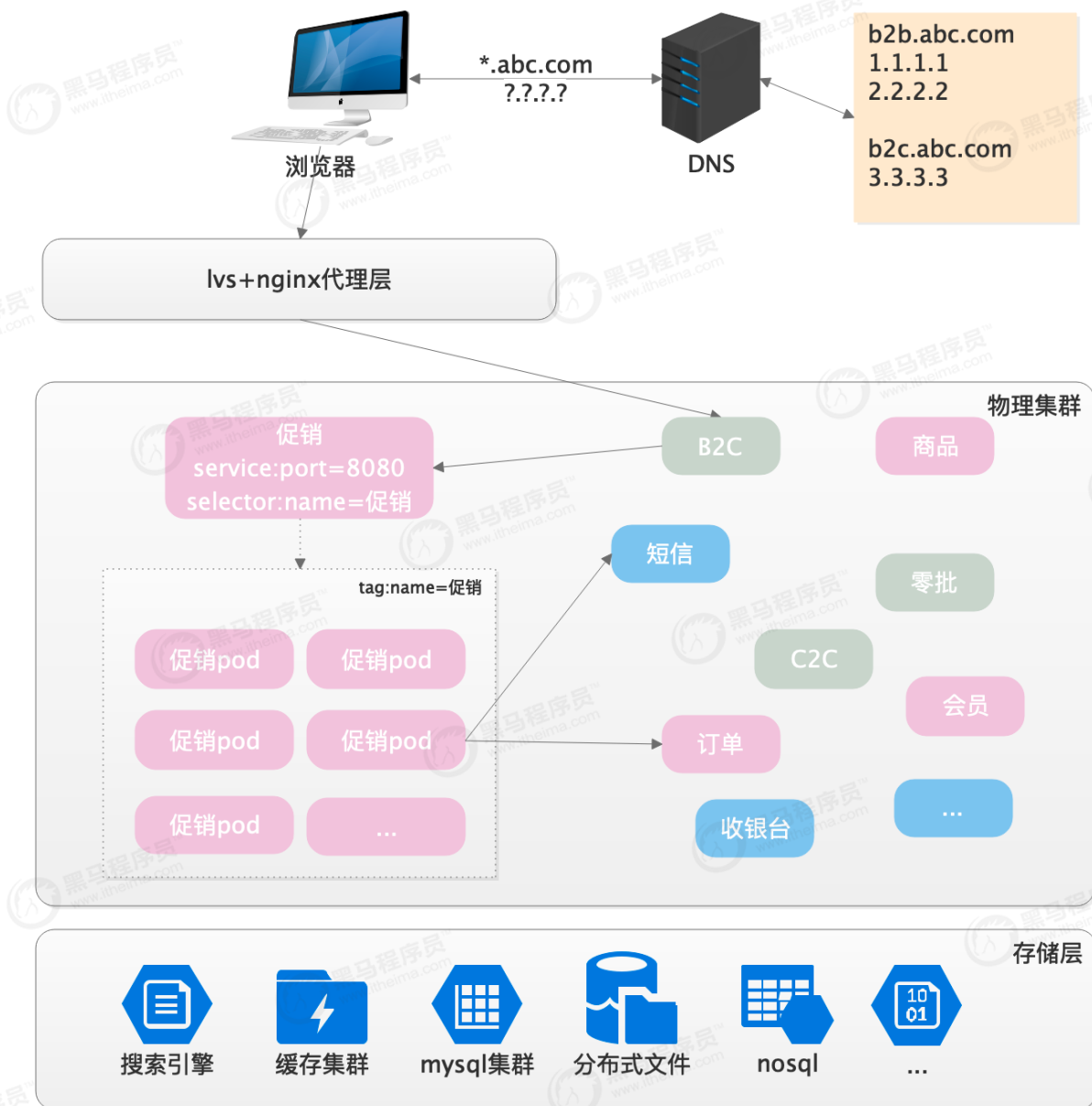
接口约束：系统增多，各个服务接口的规范化日益重要，要求有统一的服务接口规范，推动企业消息总线的建设

跨服务令牌：借助oauth2等手段，实现服务之间的权限认证

2.15 容器化

针对中台化的建设及微服务数量的飙升，部署和运维支撑同步进行着变革。面临微服务的快速部署，资源的弹性伸缩等挑战，容器化与云被推进。

案例：成百上千的服务数量庞大、大促期间某些微服务的临时扩容。



1) 方案

虚拟化：vm方案，Openstack，Vmware，VirtualBox

容器化：docker

编排：swarm，k8s，k3s

云化：容器化解决了资源的快速伸缩，但仍需要企业自备大量预备资源。推动私有云到企业云进化

2) 特点

资源预估：注意资源的回收，降低资源闲置和浪费，例如大促结束后要及时回收。

运维要求：需要运维层面的高度支撑，门槛比较高

预估风险：云瘫痪的故障造成的损失不可估量，（openstack垮掉的故事案例）

3 架构总结

1. 知行合一，做之前，先考虑意义
2. 原生优于定制，约定大于配置
3. 什么都是，最后会沦落到什么都不是
4. 控制技术欲，不要瞎折腾
5. 留下扩展，但不要想到100年后
6. 没有最好的，只有最合适的
7. 够用就好，玩的越花，风险越大
8. 简约最美