

流量从0到10亿，技术架构演进之路

快狗打车CTO-沈剑

关于-我

- 前百度 - 高级工程师
- 58同城 - 高级架构师，技术委员会主席，技术学院优秀讲师
- 58到家 - 高级技术总监，技术委员会主席
- 快狗打车（原58速运） - CTO
- “架构师之路”作者，深夜写写技术文章
- 本质：技术人一枚



互联网分层架构技术迭代 演进历程

分层架构演进与优化

- 小流量站点架构
- 中等规模站点架构
- 大流量站点架构
- 更大流量？
- XXOO！

好的架构是进化来的
不是设计来的

如何进化？

找到和解决主要矛盾！

(1) 小流量

需求是什么

- 需求

- (1) 有个能看的见的网站就踏实了

- 特点

- (1) 请求量低 (< 10w)

- (2) 数据量小 (< 10w)

- (3) 代码量小

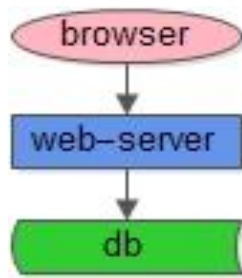
- (4) 1台机器

请求量小于10w意味着什么？

实践：如何做容量预估？

架构抽象-ALL IN ONE

- 架构图
- 架构特点
 - (1) 单机系统 (all in one)
 - (2) 程序耦合 (all in one)
 - (3) 逻辑核心是CURD



架构选型的弯路

- 选择：windows , iis , sql-server , C#
- **缘何走了“微软技术体系”这条路？**

如果重来，我们会怎么选？

创业优秀实践：LAMP

主要矛盾：CURD频繁出错

如何解决CURD频繁出错？

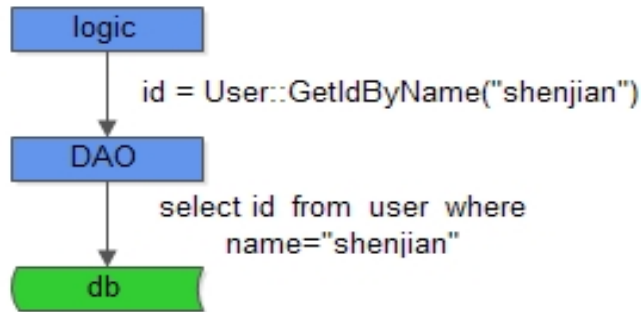
技术

- **DAO**

- (1) Data Access Object
- (2) 像访问对象一样访问数据

- **ORM**

- (1) Object Relation Mapping
- (2) 简化数据库查询过程



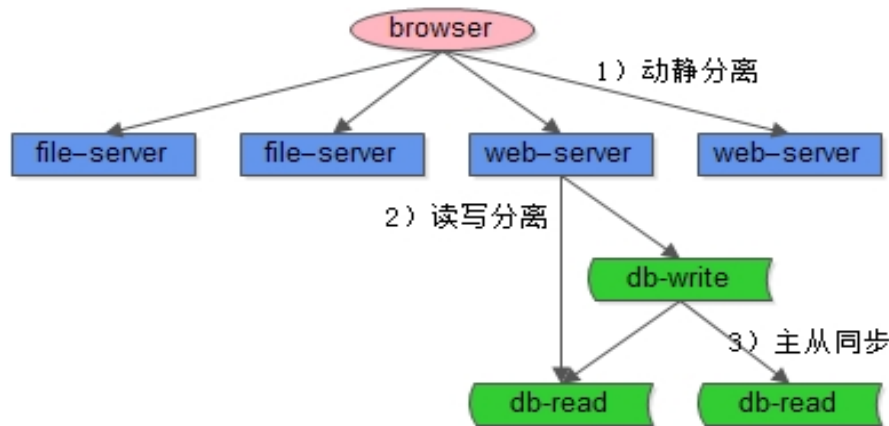
(2) 中等规模

需求是什么

- 需求
 - (1) 网站能够正常访问
 - (2) 如果访问速度能快点就最好了
- 特点
 - (1) 压力导致经常宕机
 - (2) 数据库成为瓶颈啦
 - (3) 人多的时候访问会卡
 - (4) 10+台机器

架构抽象-分布式

- 架构图
- 架构特点
 - (1) 分布式系统
 - (2) 动静分离
 - (3) 读写分离 (主从同步)



动静分离

主从同步+读写分离

主要矛盾：

站点耦合+读写延时

如何解耦？如何缓解延时？

业务-垂直拆分



武汉 切换城市

武汉 [切换城市]



多云 18 ~ 9°C

中度污染



请输入类别名称或关键字

租房子 兼职 求职 销售 电动车

首页

招聘

租房

二手房

二手车

二手市场

武汉房产

房屋出租

整租/合租

房屋求租

跳蚤市场

二手手机 苹果 / 三星 / 小米

台式电脑 显示器 / 外设

笔记本 macbook / 联想

所有职位

最新职位

名企专区

NEW

☐ 五险一金 ☐ 提供食宿 ☐ 年底双薪 ☐ 交通补助 ☐ 周末双休 ☒ 帮帮在线

公司行业 公司性质 工作经验 薪资 学历 发布时间

职业技术培训中心学费介

职业技术培训中心特色与优势,职业技术培训...

北大青鸟职业技能培训北

北大青鸟职业技能培训中国IT教育领航者,专...

☐ 诚聘前台迎宾 [2图] 顶

武汉市月桂休闲足疗有限公司 验

☐ 迎宾/接待

武汉天子问道旅游酒店管理咨询有限... 验

1 台式机/配件 (重选类别)

* 发布人 ☒ 个人转让 ☐ 商家转让

* 类别 类别

* 新旧 新旧

* 转让价格 元

物品照片

图片上传

手机传图

第-

迎宾/接待

武汉天子问道旅游酒店管理咨询有限公司硚口分店

公司规模: 1000人以上

公司性质: 国有

入驻时长: 39天

薪资待遇: 2000-3000元

学历要求: 不限

工作年限: 不限,可接收应届生

架构-垂直拆分

- 架构图

- 架构特点

(1) 站点垂直划分

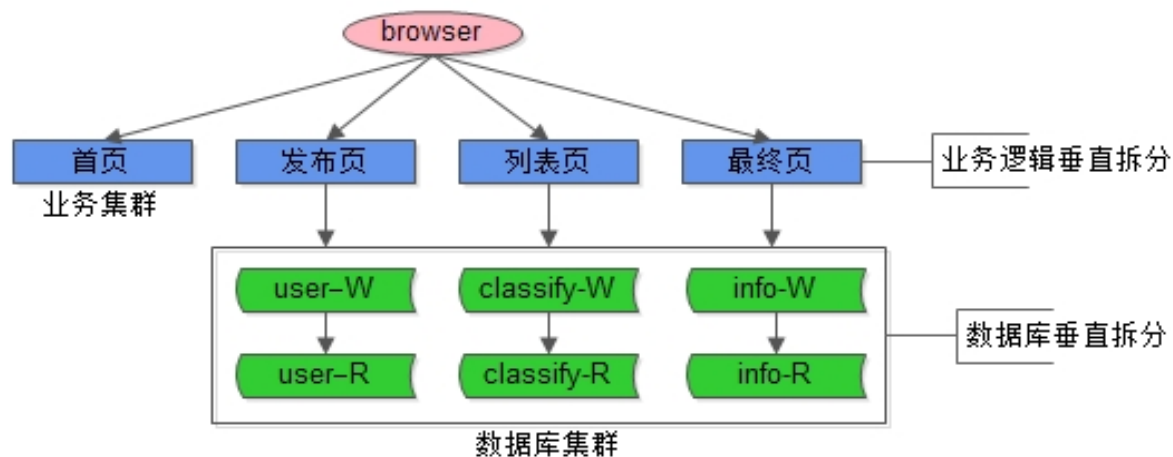
(2) 数据库垂直划分

(3) 代码垂直划分

- 主要矛盾缓解

(1) 站点耦合 => 解耦

(2) 读写延时 => 解耦



(3) 大流量

需求是什么

- 需求
 - (1) 垂直业务也不能挂
 - (2) 业务爆发-快速实现
 - (3) 业务依赖-子系统依赖关系复杂
- 特点
 - (1) 站点数激增
 - (2) 数据量激增
 - (4) 100台机器

架构抽象-高可用

- 架构图

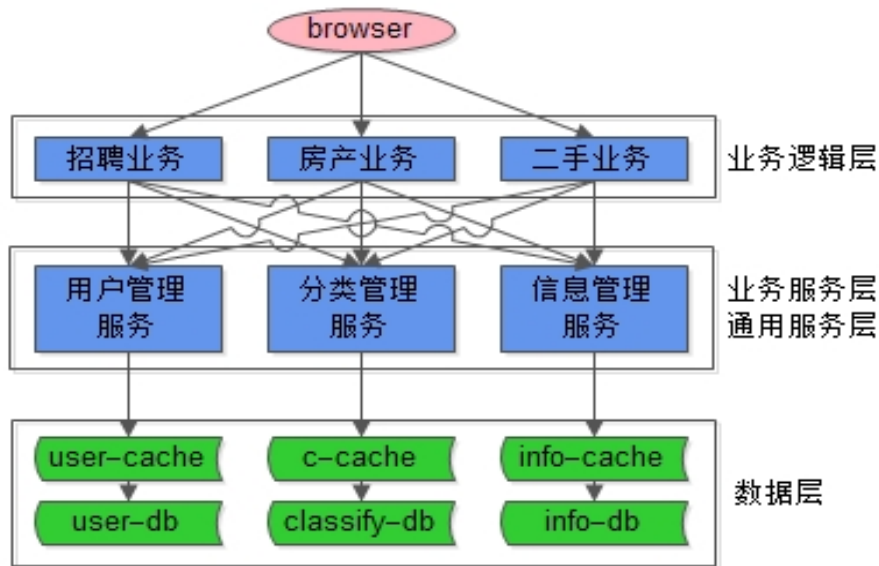
- 架构特点

(1) 进一步垂直拆分

(2) 分层抽象

(3) 服务化

(4) **水平拆分**



矛盾点：高可用

高可用的思路是
冗余+故障转移

矛盾点：高性能，高并发

如何做到 “无限性能” ？

无限性能的思路是
加机器就能扩展

矛盾点：臃肿，耦合

为什么要服务化？

互联网典型服务化架构

- 联网典型高可用架构

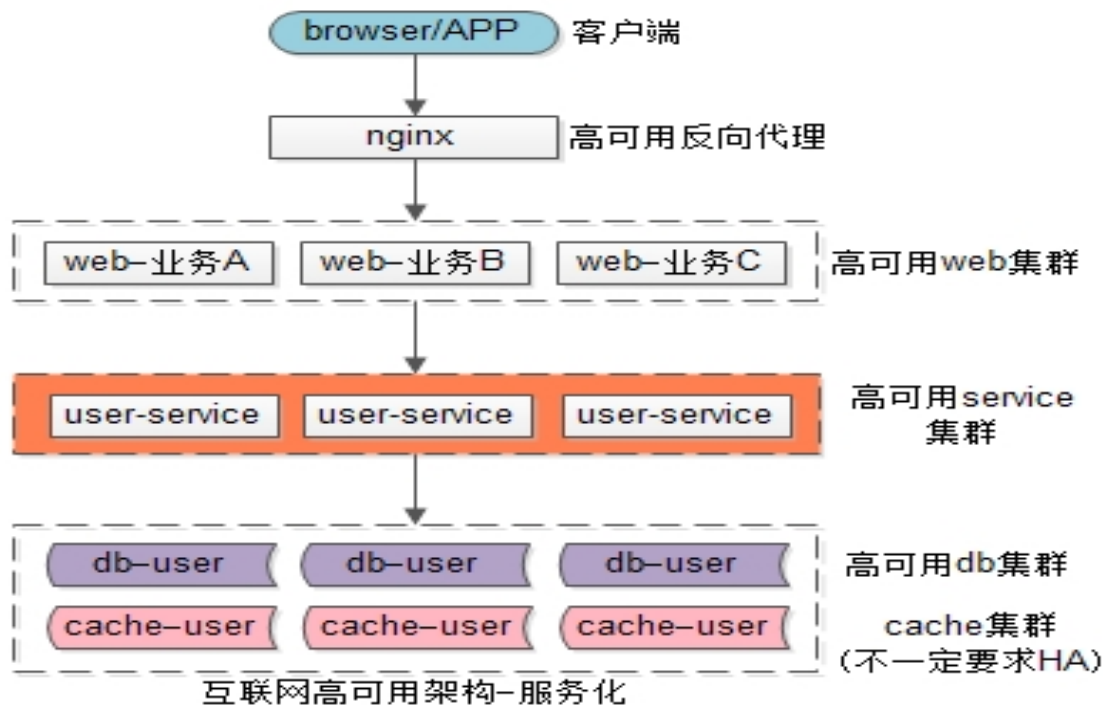
(1) 端

(2) 反向代理

(3) 应用

(4) 服务

(5) 数据



服务化解决什么问题

- 向上层屏蔽底层细节，调用方很爽
- 复用性
- 接耦合（系统、数据库）
- 专注性
- SQL质量得到保证
- 提供无限性能（有限服务）

(4) 更大流量

需求是什么

- 需求

- (1) 用户量、数据量、并发量暴增

- (2) 业务量暴增

- (3) 迭代，敏捷

- 特点

- (1) 自动化

- (2) 平台化

- (3) 综合治理

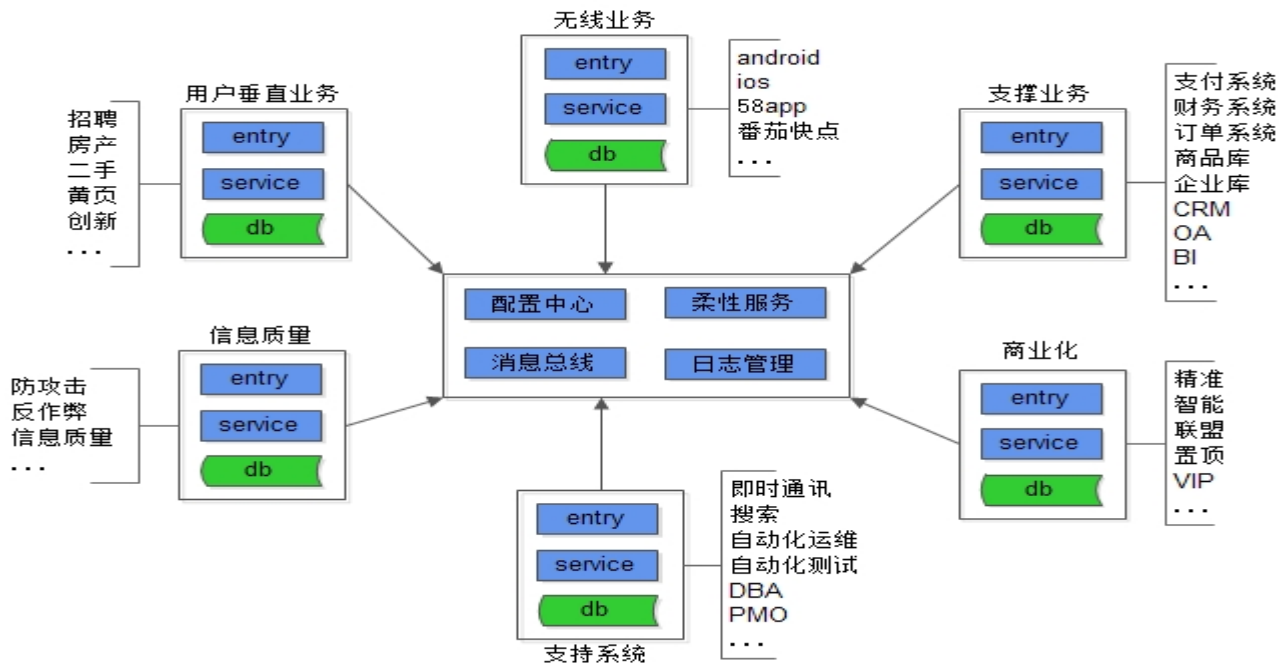
- (4) 1000台机器

复杂性矛盾：

架构变成蜘蛛网

架构抽象-进一步解耦

- 架构图
- 架构特点
 - (1) 配置中心
 - (2) 消息总线



配置中心

消息总线

多维分层

迭代思路

通用痛点抽象一层

微服务架构其他最佳实践

统一的服务框架

技术-开源的开发框架

- 如何降低**站点**开发成本？

(1) web框架

(2) <https://github.com/58code/Argo>

- 如何降低**服务**开发成本？

(1) 服务框架

(2) <https://github.com/58code/Gaea>

统一的数据访问

总结

- 容量预估
- 三大分离：动静，读写，前台与后台
- 高可用，冗余+故障自动转移
- 扩展性，高并发，加机器就能扩容
- 为什么要服务化
- 配置中心，逻辑解耦，物理不解耦
- 消息总线，逻辑解耦，物理解耦
- 多维分层，通用痛点抽象一层

Q&A

欢迎交流

谢谢！



讨论：秒杀实践