

架构专栏特别放送！“华仔，放学别走”第2期  
2018-06-01 柯鹏 & 李运华

更多一手资源请添加QQ/微信1182316662



各位同学，晚上好，我是架构专栏的编辑Shawn。今天又到周五啦，没错，我又出来送福利了[捂脸]。

“华仔，放学别走”第1期不知道你看没有，华仔回答了关于知识分享、理论与实践、专栏学习方法、推荐的参考书等几个问题，希望你从中能够有所收获。今天是“华仔，放学别走”第2期，继续回答你所关注的问题，然后展示出08 ~ 13期被选中的精选留言，并给留言被选中的同学送出价值68元的专栏阅码。话不多说，开始今天的问答环节。

Shawn：有做公司架构/网站架构/App架构的同学，这个专栏能帮助到他们吗？

华仔：有的同学在学习了一段时间后跟我留言交流，说感觉专栏的内容好像比较适合做互联网后台架构，不太适合企业应用、客户端这类系统。其实这是一个误解，我之所以在前面花费很大篇幅来讲架构设计的目的、架构设计原则、架构设计流程等看起来比较偏理论的内容，而没有一上来就讲异地多活、高性能架构之类的怎么做，原因就在于这是一套完整的架构设计理论体系，不管是企业应用，还是客户端应用，都可以按照这个设计理论体系去操作。我以手机App为例，首先，我们分析一下App的复杂度主要来源是什么？通常情况下，App的主要复杂度就是可扩展，因为要不断地开发新的需求；高性能和高可用也涉及，高性能主要和用户体验有关；高可用主要是减少崩溃。其次，再看App的架构需要遵循架构设计原则么？答案是肯定需要。刚开始为了业务快速开发，可能用“原生+H5”混合架构；后来业务发展，功能更复杂了，H5可能难以满足体验，架构又需要演进到“纯原生”；如果业务再发展，规模太庞大，则架构又可能需要演进到“组件化、容器化”。以上通过手机App的为例说明这套架构设计理论是通用的，有兴趣的同学可以按照这种方式分析一下企业应用，会发现这套理论也是适应的。

Shawn：讲讲你总结“架构设计三原则”的过程吧？

华仔：“架构设计三原则”是综合各方面的信息和思考得来的。首先是我自己的经验，包括成功的经验和失败的教训；其次是分析了很多业界的架构演讲和技术发展历史；第三是看了一些关于技术本质的书籍而受到的启发，例如《技术的本质》《系统之美》等。其实最初整理的架构设计原则有10多条，但我觉得10多条太多了，不聚焦也不利于理解，因此去芜存菁，最终得到了“架构设计三原则”，这三个原则是最重要也是最核心的。

如下是我原来整理的设计原则：可以看到一共有14条：

架构设计原则：

- 1) 合适优于业界领先：可以参考淘宝、但不要照搬淘宝
- 2) 演化优于过度设计：不要设计过于超前的方案，但演化不要推倒重来
- 3) 简单优于复杂：轮询大部分情况下都是很好的设计，不确定就穷举，不明确就轮询
- 4) 重构优于重写：风险控制、经验传承、成本分散
- 5) 硬件优于人工：能够用硬件解决的，不要用人工去解决，例如换SSD，换更强的机器
- 6) 专注优于全面：SRP原则，一个系统只关注一个事情，JAE的例子
- 7) 开放优于封闭：亚马逊的例子，SOA、微服务
- 8) 能用优于完美
- 9) 重用优于自研？
- 10) 业务优于技术：docker很火，我们是否要引入？
- 11) 存储优于运算：存储设计是架构设计的关键
- 12) 技术优于流程：不要试图提升人的能力来保证质量，不要试图投入大量的测试来保证质量，而是尽量做到出问题能够快速发现和处理
- 13) 分布优于集中
- 14) 优化优于重构

Shawn：“PPT架构师”的口头禅是“细节不讨论”，一个优秀的架构师，需要对细节有多少考虑呢？

更多一手资源请添加QQ/微信1182316662

华仔：这是一个非常好的问题，也是很多同学困惑的问题。我分享一下我的做法，以我学习Elasticsearch为例，具体的做法是：

- 1. 搭建一个单机伪集群，搭建完成后看看安装路径下的文件和目录，看看配置文件有哪些配置项，不同的配置项会有什么样的影响。
- 2. 执行常用的操作，例如创建索引，插入、删除、查询文档，查看一下各种输出。
- 3. 研究其基本原理，例如索引、分片、副本等，研究的时候要多思考，例如索引应该如何建，分片数量和副本数量对系统有什么影响等。
- 4. 和其他类似系统对比，例如Solr、Sphinx，研究其优点、缺点、适用场景。
- 5. 模拟一个案例看看怎么应用。例如，假设我用Elasticsearch来存储淘宝的商品信息，我应该如何设计索引和分片。
- 6. 查看业界使用的案例，思考一下别人为何这么用；看看别人测试的结果，大概了解性能范围。
- 7. 如果某部分特别有兴趣或者很关键，可能去看源码，例如Elasticsearch的选举算法（我目前还没看^\_^）。
- 8. 如果确定要引入，会进行性能和可用性测试。

这样一套组合拳下来，基本上能够满足在架构设计时进行选型判断，而且花费的时间也不多。我并不建议拿到一个系统一开始就去读源码，效率太低，而且效果也不好。

Shawn：谈谈架构师沟通能力的重要性吧？

华仔：架构师是业务和技术之间的桥梁，同时通常情况下还会确定整体项目的步骤。因此，架构师的沟通能力非常重要，既要说得动老板，让老板支持自己的设计决定；又要镇得住技术人员，让技术人员信服自己的设计选择；同时还要能够理解业务，结合业务不同发展阶段设计合适的架构，所以也要参与产品和项目决策。由于架构设计过程中存在很多判断和选择，而且不一定都有明确量化的标准，因此不同的人有不同的看法是普遍情况。这种情况下架构师既需要专业能力过硬，又需要具备良好的沟通技巧，才能促使业务、项目、技术三方达成一致。

当然，架构师的核心能力还是技术能力，过硬的技术才是良好沟通的基础，否则单纯靠沟通技巧甚至花言巧语，一次两次可能奏效，但后面被打脸打多了，也就没人信任了。

Shawn：有同学留言说，给企业做项目，甲方会不顾业务需要，只要是业界流行的技术就要求在项目中采用，这种情况下怎样才能符合“架构设计三原则”？

华仔：首先，业务第一，先把订单签下来，才有后面的架构设计，如果硬要说甲方的要求不合理，不满足“架构设计三原则”，结果订单都拿不到，那是没有意义的。其次，这种情况我把它归为“架构约束”，即这不是架构师能够选择的，而是架构师必须遵守的，因此这里不需要使用“架构设计三原则”来判断。第三，这种情况下，架构师还是可以应用“架构设计三原则”来指导架构设计，比如说客户要求采用Docker，Docker的网络模式有5种，host模式使用起来比bridge模式简单，那我们就用host模式；如果客户再要求需要对Docker进行统一管理，那我们是自己研发Docker管理平台，还是直接用Kubernetes呢？按照简单原则来说，肯定用Kubernetes了。

通过这个示例也可以看出，“架构设计三原则”主要是指架构师在选择和判断时采取的指导原则；但如果是架构的基本需求或者约束必须被满足时，架构师此时的选择是采取什么样的方案能够更好的满足这些需求和约束。

留言精选



更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

当时也是拍脑袋想到为何不用业界流行的框架

重构我们的系统，其实没有最好的框架，只有合适的框架，只要能够简单的解决系统面临的业务复杂度，架构组优先会选择公司现有的框架

另外提到架构演绎，我的看法应该是偿还技术债务，公司目前推行的是敏捷开发，敏捷开发的價值就是快速可靠的持续交付，往往 team 实践时候优先考虑如何在现有框架基础上快速实现业务需求

长期以往一个组件的功能就非常复杂了，功能上容易牵一发而动全身，所以这时候不得不让 team 停顿下，解决现有的技术债务，从而让复杂的组件从功能上解耦

引自：从0开始学架构

08 | 架构设计三原则



识别二维码打开原文  
「极客时间」App

华仔：有个懂技术的好老大是一件多么幸福的事情：)

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

Michael

写于 2018/05/18

老师您好，有几个问题

1. 文中提到的阿里腾讯最初是用简单的方案这跟当时的环境有关，必须快速开发出产品，可能钱也没那么多

如果当时是微软这样的巨无霸公司，他们有市场也有钱了，那是不是可以一上来就设计出淘宝 2.0 或者淘宝 3.0

所以简单，合适，是被环境逼出来的，必须这么做？

2. 设计三原则中的演化该如何考虑，比如当时不好预测，那就等以后再重构演化，可是未来业务量大了，可能会大规模重新设计，可能会浪费大量人力物力，甚至错失发布产品的好实际，那么最初设计演化的时候该如何把握好这个尺寸呢？

引自：从0开始学架构  
09 | 架构设计原则案例

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

识别二维码打开原文  
「极客时间」 App



华仔：有钱也不能任性，微软95年也不可能开发出Windows 10操作系统，业务量大了重构甚至重写那是自然而然的，不会浪费也不会导致错失产品机会，Windows、Android、淘宝、QQ都是这么过来的。

# 空档滑行

写于 2018/05/20

说下之前改造的一个系统，当时是这个系统从其他系统同步数据，经过一整套流程后将数据拆解到本地库的各个业务表中。

原来的系统是一个单机多线程程序，到了大促的时候延时非常厉害，因为马上又要大促了，首先想到的是扩展性的问题，于是就做了无状态服务拆分，可以横向扩展。在这过程成因为要控制单个同步实体任务的并发在处理幂等性上也花很大的功夫。做完这个后，又对非关键流程做了消息解耦，提高主流程的处理速度。

系统上线后运行稳定，但是到了大促当日发现速度提升很有限，将机器扩展到 10 台速度只

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

提升了 2~3 倍，而且数据库和应用压力都很小。

后来分析下来瓶颈竟然是在数据库中间件上。其他系统大量的慢 sql 导致中间件处排队严重。

现在想起来，其实还是改造前优先级没搞清楚，根据之前的经验就把问题归结到扩展性上，投入了大部分精力在扩展性和可用性上，而没有对原来单机系统的性能做完整的测试和评估。其实当时做一个单机程序的压测大概就可以判断出问题所在（改造过程中也发现老的程序确实有 bug）。还有就是高性能的优先级远大于扩展性和可用性，因为这个系统重启的影响非常小。改造的时候过于理想化的，想做一个各个方面均衡完善的架构

引自：从0开始学架构

10 | 架构设计流程：识别复杂度

识别二维码打开原文  
「极客时间」App



更多一手资源请添加QQ/微信1182316662



更多一手资源请添加QQ/微信1182316662

## bluefantasy

写于 2018/05/22

老师好，个人觉得像前浪微博这种场景使用 Kafka,Rabbitmq 会比 Netty+Mysql 自研好很多。主要原因是这个场景对实时性要求还是比较高的（一般采取消息队列主动推送模式）。开源的消息队列都有对消费者的推送模式。自研的话，如果采用消息推送模式，消息队列服务需要在服务端记录所有消费者的状态信息，还要考虑各种异常和消息确认，实现起来应该是很复杂。一般公司根本没有这个技术实力。个人见解，希望得到老师的回复。

引自：从0开始学架构

11 | 架构设计流程：设计备选方案

识别二维码打开原文  
「极客时间」App



更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

## 东

写于 2018/05/27

案例很典型，所在项目，先选了 3，1.0 上线后效果不错，后期业务扩展，投入跟不上，3 的缺点不断暴露，到后来大家就在吐槽为啥要造轮子。开始否决 3，重构，选择了 1，运维话语权弱，被忽略了。至于为啥不选 2，就是面子上过不去，拿不出手。项目不光是为了业务，也为了架构师，领导的面子，被拿来和公司内其他项目做横向比较时，比较好吹。至于运维的哥们，也乐意学些新东西，提升自我价值。所以，选择 1 大家都开心，除了项目的投入变大

引自：从0开始学架构

12 | 架构设计流程：评估和选择备选方案

识别二维码打开原文  
「极客时间」App



更多一手资源请添加QQ/微信1182316662



## ant

写于 2018/05/26

很有幸我们现在的架构师就是 PPT 架构师，我觉得他的优点就是懂了很多的概念，能说到，可以忽悠住老板。缺点也很明显，就是他知道的都不是很深，比如曾经我们的搜索引擎原型，他并不能说出 ES 和 solr 的优缺点（当然我也不知道，平时用 solr 多点），最后我们选了 ES，他给的原因就是朋友说的 ES 比 solr 好，后面搜索这里就交给我来搞了。我们是互联网项目，在重构的项目的时候他选择了 jpa，这就导致变化需求的时候，查询这块比较麻烦，不灵活。

其实就像前面说的，每个技术存在就是合理的，只是每个有每个技术的使用点，架构师应该对常见的技术栈原理非常清楚，知道什么时候应该使用什么技术。

我理解的 PPT 架构师的特点就是知识点多，知道概念，能忽悠住老板。缺点就是技术栈不纯

更多一手资源请添加QQ/微信1182316662

滴，特别是细节上。我觉得架构师应该帮助员工成长，而不是遇到问题就说这个问题我没遇到过，你上网搜索下解决方案。

初次留言，欢迎板砖

引自：从0开始学架构  
13 | 架构设计流程：详细方案设计

识别二维码打开原文  
「极客时间」App



华仔：架构师确实需要在技术广度和技术深度两方面都要兼顾，但如何把握技术深度这个“度”，不同架构师有不同的理解，但千万不能说“细节不讨论”“你上网搜”，这样会没有技术公信力。

最后，再次恭喜@Tony、@Michael、@空档滑行、@bluefantasy、@东、@ant，也感谢写下留言的每位同学。欢迎你在这期“华仔，放学别走”留下你的问题，业务、职场、职业规划等不限主题，可以和华仔一起聊聊专栏以外的话题。



爱吃技术的◆◆

企业高速发展过程中，技术总是短期内被高估，长期被低估。一位15年IT老兵的切身感受！  
作者回复

我理解是短期被牺牲，长期被低估◆◆◆◆  
张国胜

2018-06-02

2018-06-02

更多一手资源请添加QQ/微信1182316662

	2018-06-29
之前在公司遇到这个问题，拿演化当时来说，其实将之没有架构就写开干，演化你写啥就写啥，你的目的是，演化要什么就演化，能看到一定程度的将来，然后从多种设计方案中选择最基本的那一个，就得到了最小子集。这事后就演化的话，是的不现实和基础，如果真从早期开始，而不像现在作为演化。	
作者回复	
是的，演化优于过度设计不是说不要设计，而是不要过度设计	2018-06-30
Panda	
目前公司的项目开发量不大，项目比较复杂，逻辑复杂度高，多个项目耦合强，请老师分享一下这种情况下的架构 应该怎么做	2018-06-04
作者回复	
可扩展章节有讨论	2018-06-06
东	
忽然发现上了热门，有种被翻牌子的感觉!	2018-06-04

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662



更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662