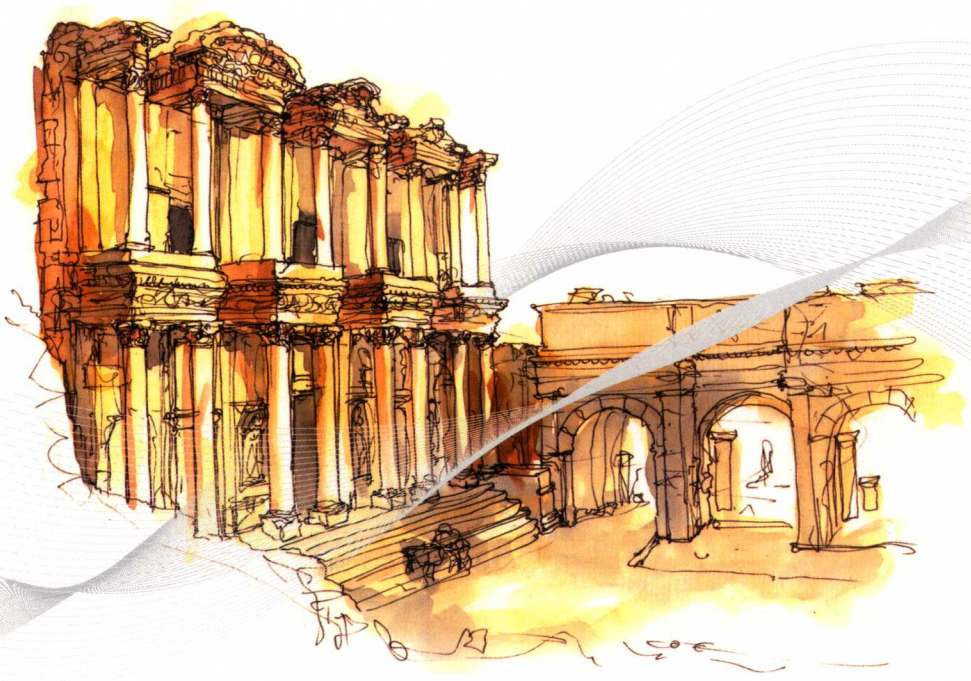
Broadview®
www.broadview.com.cn



非外借

大型网站
技术架构演进与性能优化

许令波◎著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

大型网站
技术架构演进与性能优化

许令波◎著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书从一名亲历者的角度，阐述了一个网站在业务量飞速发展的过程中所遇到的技术转型等各种问题及解决思路。从技术发展上看，网站经历了 Web 应用系统从分布式、无线多端、中台到国际化的改造；在解决大流量问题的方向上，涉及了从端的优化，到管道、服务端，甚至基础环境优化的各个层面。

书中总结的宝贵经验教训可以帮助读者了解当网站遇到类似问题时，应如何思考不同的解决思路、为什么要这样做、并最终选择合适的方案。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

大型网站技术架构演进与性能优化 / 许令波著. —北京：电子工业出版社，2018.7
ISBN 978-7-121-34135-9

Ⅰ. ①大… Ⅱ. ①许… Ⅲ. ①网站－开发 Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字(2018)第 087106 号

责任编辑：刘　皎
印　　刷：北京画中画印刷有限公司
装　　订：北京画中画印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：720×1000　1/16　印张：13.5　字数：284 千字
版　　次：2018 年 7 月第 1 版
印　　次：2018 年 7 月第 1 次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

好评袭来

君山经历了淘宝网发展速度和架构变化最快的时代，这是一个机会和挑战并存的时代，许许多多无法用常理理解的需求不断涌现，许许多多从未遇见过的问题横在面前，许许多多创新的解法横空出世！君山把传统的软件工程开发理念和新机遇下的技术创新相结合，在性能优化领域不断创新：小到字节码层面的优化、大到架构上的重建——他的探索工作在淘宝网的技术发展史上留下了痕迹。

君山做事情有几个特点：一是能把技术和业务相结合，在处理业务需求和问题时轻车熟路，在处理完业务需求的同时还会带来技术上的创新；二是善于推动技术创新落地，用自己的实践诠释了“创新只有被人使用、在业界形成潮流才算是真正的创新”这句话；三是善于总结思考，他每次都把技术和业务上遇到的问题和解法总结下来，并乐于分享，让团队共同成长！

——阿里巴巴研究员　小邪

做技术做到后期才会发现写代码并不是全部。随着业务的快速迭代，对系统的架构演进和相关技术的权衡会变得越来越重要，在不同的阶段会有不同的取舍。特别是大型系统，除了要考虑技术，还要考虑相匹配的组织架构、工程文化等因素——这些挑战是很难通过亲历来获取的，毕竟成功的大型系统不算太多。

作者曾是淘宝网一线的技术专家，亲身经历了淘宝网业务飞速增长的过程，并将其中的经验和学习的过程记录下来，完整地为我们展现了一个初级系统在演化成一个全球分布式的系统的过程中，从语言选择、分布式框架改造、平台化演进、系统优化到稳定性建设等关键过程的思考，内容翔实可信。在这些最佳实践中，技术点也许并不是最重要的，读者可收获多维度的启发和共鸣，推荐阅读！

——阿里云研究员　褚霸

一家伟大的互联网企业一般都离不开高超技术的支撑，而高超技术的养成又离不开每天迎面而来的各种挑战。本书作者有幸经历了淘宝网这些年的技术巨变，碰到了无数的问题，积攒了很多并发架构设计和性能优化的经验。好的架构是一个系统的根本，好的性能是一个系统稳定运行的保证，本书应该可以给大家带来不一样的收获。

——PerfMa CEO　你假笨（寒泉子）

针对C端用户的互联网业务是爆发式的、井喷式的，其带来的用户流量压力和对计算能力的要求也是非常惊人的，如何利用廉价的架构设计来部署分布式服务以应对亿级流量的场景是个非常严峻的问题。《大型网站技术架构演进与性能优化》一书讲解了高可用架构演化的进程，并提供了互联网架构性能优化的方法。正所谓互联网技术唯“快”不破——解决了性能问题，其他问题也就迎刃而解。如果你的业务正处于流量并发暴增与系统架构变革的十字路口，那么本书恰好就是你的“菜”。这是一本关于互联网高并发架构设计的优秀书籍，它从各角度剖析系统设计的演化与优化，循序渐进地将一系列复杂问题阐述得清晰、简单、易懂，是一本理论与实践相结合的实用书籍。

——《分布式服务架构：原理、设计与实战》、《可伸缩服务架构：框架与中间件》
作者　李艳鹏

对于一个高并发大流量网站的架构师而言，你的系统到底能够承受多高的并发、多大的流量，只有在你的系统经历了更高的并发、更大的流量以后才能知道。事前再多的设计、评审、测试、预演也只能让你相信，而不能让你知道。淘宝网作为全球最大的电子商务网站，每年的双 11 都会承受这个星球上可能是最大的并发访问压力，

试读结束，需要全本PDF请购买 www.ertongbook.com

那么淘宝的技术人员遇到了哪些挑战？做了哪些工作？感谢这本《大型网站技术架构演进与性能优化》，让我们一窥究竟。

——《大型网站技术架构：核心原理与案例分析》作者　李智慧

君山老师曾多次出席技术大会 SDCC 并担任讲师及出品人，为技术总监、架构师等参会者带去了很多干货的分享。实践出真知，任何脱离实际工作的讨论无疑在浪费宝贵的时间。作者在淘宝网经历了 Web 应用系统从分布式、无线多端、中台到国际化的改造；在解决大流量问题的方向上，积累了很多从端的优化，到管道、服务端，甚至基础环境优化的经验，这些助力他真正成为我们技术社区的明星专家，相信此书肯定会给广大的技术开发者带来一线的知识。

——CSDN 主编　钱曙光

前言

从 1 亿到 50 亿的技术之路

从 2009 年到 2016 年，笔者非常幸运地经历了网站 PV 从 1 亿到 50 亿的飞速发展历程，在此过程中积累了一些大流量高并发网站架构设计和优化的经验。从技术发展来看，笔者经历了 Web 应用系统从分布式、无线多端、中台到国际化的改造；在解决大流量问题的方向上，积累了很多从端的优化到管道到服务端甚至到基础环境优化的经验。现在您手头这本书所介绍的内容，大部分是笔者看到、学到的，是亲身参与和实践的经验。

本书要表达的内容并不是简单罗列所做过的事情，而主要是帮助读者了解当网站遇到类似问题时，应如何思考不同的解决思路、为什么要这样做、如何做出最终的方案选择……其实每种架构的选择必然有它专属的现实场景，因此本书涉及的这些话题也不一定就是最完美的解决方案。但，我希望本书的分享能启发大家在解决类似问题时的思考和判断。

一、架构演进之路

本书分成两个部分。第一部分主要介绍整个网站由于业务发展所经历的几次主要的架构演进，包括：从 PHP 到 Java 的改造、分布式改造、无线化改造、中台的改造、国际化改造。第二部分主要介绍如何从不同的层次解决整个网站在大流量情况下遇到

的性能瓶颈，包括端和管道的优化、应用层代码级优化、应用架构的优化、端到端的全链路优化。最后介绍在做架构和性能优化的过程中必须面对的稳定性问题——如何体系化地解决网站的稳定性，这是非常关键的。

阶段一 从 PHP 到 Java

很多网站早期都是基于 Linux+Apache+MySQL+PHP 架构的网站，从当时来看，这种非常流行的个人网站架构的确也非常匹配当时的发展状态。PHP 语言的特性是快速发布，从页面渲染到数据库访问，均可以在一个页面里全部搞定。

即使放到今天，这种架构仍然还有很多人在用，它的优点就是非常简单高效，但缺点也非常明显：扩展性和分布式不好，不适合企业级的、复杂业务逻辑的大规模协同开发。

随着网站的发展，大家觉得应该将 PHP 切换到 Java。为什么要切换到 Java 语言呢？一般来说，企业选择开发语言会有如下考虑。

（1）语言本身的特性。每种语言开发出来都有它的特性和所适合的场景，像 Python、PHP 这类脚本语言非常适合快速简单的开发方式，而 Java 则比较适合构建复杂业务逻辑的企业级开发，但是开发效率会比 PHP 要差一点。

（2）程序员队伍。企业选择何种开发语言，还要看市场上的人才队伍是不是足够大，是不是有很高层次的人才。是否有高层次的人才，取决于当前的行业老大是不是也在用这种语言，比如当前的顶级互联网公司如果在用 Java，那么自然这些公司的 Java 人才比较多，这样，他们的经验可以被快速复制到其他公司中。

（3）语言所对应的工具生态是否完善。一个语言是否有生命力，要看这个语言对应的生态工具是否完善，它的社区是否活跃。我们要用到各种工具，而我们也不可能自己去写每种工具，因此，是否能方便地利用开源工具，快速提升开发效率也是非常关键的。像现在很多大公司开源了很多 Java 的中间件产品，这些中间件可以直接拿来使用，就不需要再重新开发了。

综合以上因素，电商网站选择 Java 语言作为主要的系统开发语言是非常合适的。

从 PHP 切换到 Java 后，整个网站采用 WebX+EJB+iBatis+JBoss+Oracle（后面又将 EJB 改成 Spring）的架构，但是随着业务量的不断增大，存储层的瓶颈暴露出来。为了解决存储问题，就逐渐用上了非常昂贵的 IBM 小型机、Oracle 的数据库以及 EMC 的高端存储（IOE）；并对数据库做了分库的拆分，分布式缓存（Tair）也随之诞生，分布式文件系统 TFS 开始出现，CDN 也慢慢建立了。

阶段二 分布式改造

所谓分布式改造，就是尽量让系统无状态化，或者让有状态的信息封装在一定范围内，以免限制应用的横向扩展。简单来说，就是即便某个应用的少数服务器“宕”掉，也不会影响整体业务的稳定性。

要实现应用的分布式改造必须先解决以下几个问题。

（1）把应用微服务化：即将大量粗粒度的应用逻辑拆小，做服务化改造。

（2）必须先建立分布式服务框架。必须具备分布式 RPC 框架、异步消息系统、分布式数据层、分布式文件系统和服务的发现、注册和管理。

（3）必须要解决分布式 Session 问题。

为了做业务的服务化改造，我们大量拆分了当时的业务系统，形成了商品中心、交易中心、用户中心、店铺中心。这些服务作为底层服务供上层的前台系统调用，此时的系统架构变成了如图 0.1 所示的形式。

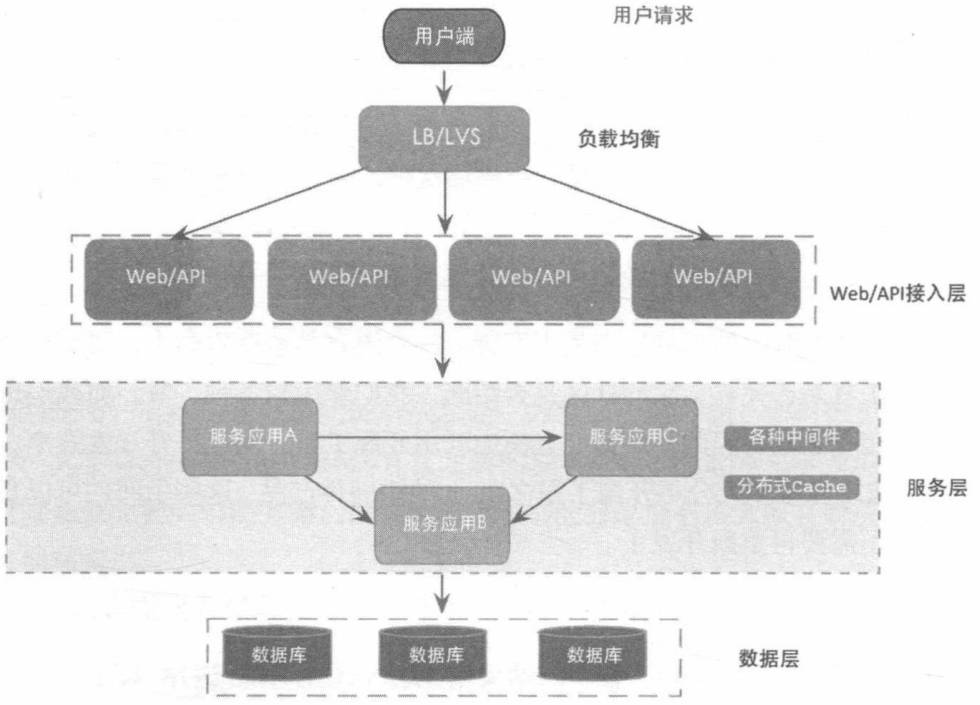


图 0.1　分布式改造后的架构

现在来看，系统的分布式改造为网站接下来 5 年的发展奠定了很好的基础，整个网站的扩展性非常好。几乎每个初创企业都必须经历一次分布式化的改造，才能为企业的长期发展奠定基础。笔者前面提及的几个重要的中间件产品和关键的分布式 Session 等技术在《深入分析 Java Web 技术内幕（修订版）》一书中有详细的介绍，感兴趣的读者可以自行去学习了解。

阶段三 无线化改造

到了 2013 年，无线技术已经非常火爆了。在此之前，无线的业务总是跟着 PC 走，基本上是 PC 做好后无线再复制一份，而且无线和 PC 还不是同一个前台系统，导致一个功能要做两遍，并且无线部门的开发人员本来就不多。这些给业务的发展带来很多问题。因此 2013 年底，启动了 All in 无线项目，目标就是用一套系统、一套架构快速支撑多端的个性化，并在服务端做了很多模块化和组件化的改造。

随着无线技术的发展，我们也开始尝试一些新技术，最具代表性的就是 Node。Node 在业界很火，前端同学非常推崇，而且它也大幅提升了前端的开发效率和体验。目前大家也多方尝试使用 Node 技术，并且用在一些业务线上。但是，从今天来看，Node 并没有达到我们想象中的发展前景。这其中有多种原因，本书的后续章节会专门介绍网站的无线化改造实践，也将分享 Node 在实际使用中的一些思考。

经过无线化改造的应用系统架构如图 0.2 所示。

阶段四 中台改造

中台这个概念早期是由美军的作战体系演化而来的，技术上的“中台”主要是指学习这种高效、灵活和强大的指挥作战体系。电商经过十几年的发展，组织已经庞大而复杂，业务不断细化拆分，也导致野蛮发展的系统越来越不可维护，开发和改造效率极低，也有很多新业务不得不重复造轮子，所以中台的目标是解决效率问题，同时降低创新成本。书中会有单独的一章介绍笔者看到、学到的中台的实践和思考。

阶段五 国际化

国际化一般会有两种思路：一种是一套原始代码部署到多个地方，各地的系统基本没有什么关联、保持相互独立，每个地方再根据本地实际情况做一些个性化的定制。一般来说，会精简原始代码，减少不必要的依赖。这种思路在一些跨国公司用得比较多，但是这个对技术要求比较低，不是我们介绍的重点。

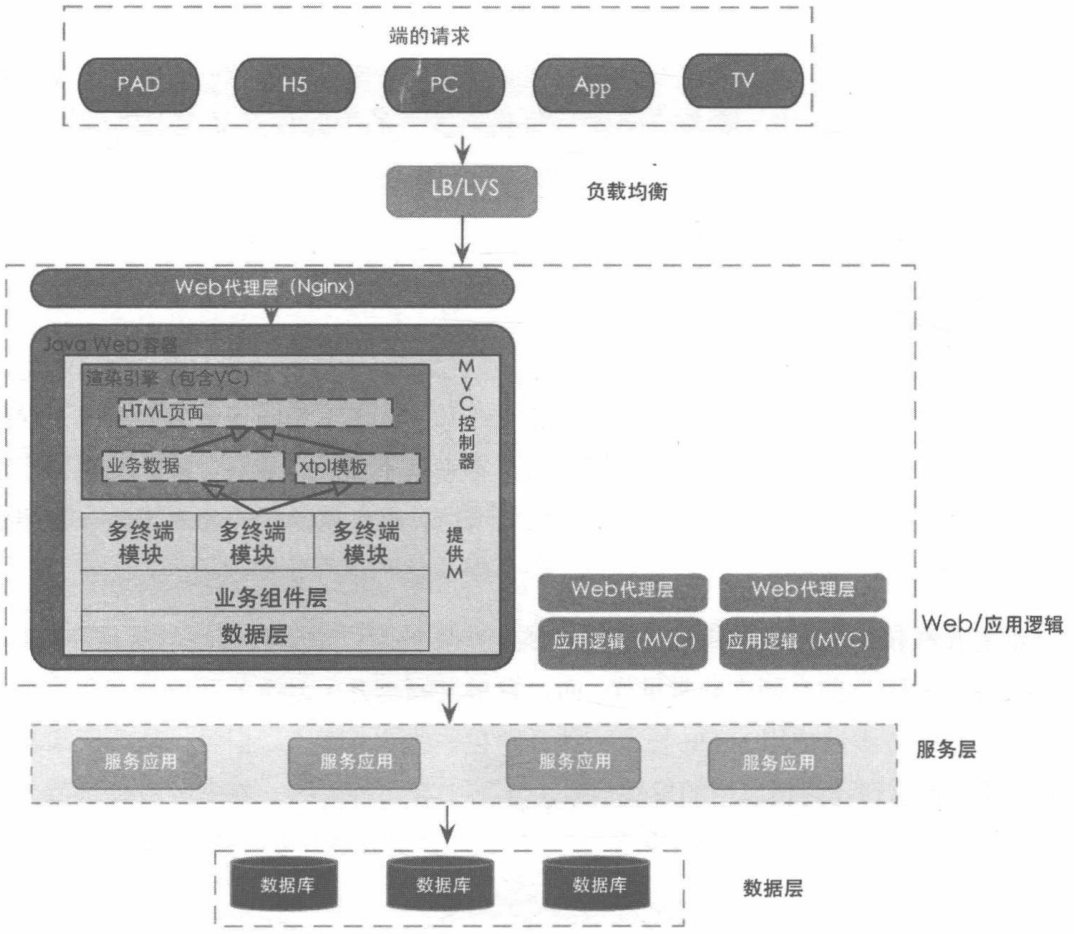


图 0.2　无线化改造后的应用系统架构

另一种思路就是我们要介绍的国际化，它主要解决如何将一套系统部署到多地的问题。一般国际化有两个发展阶段：第一个阶段是在国内实现了交易的单元化；第二个阶段是实现了中美的跨国部署。

国际化的本质仍然是要解决以下的通用问题：多语言问题、多时区问题、数据路由问题、全球数据的同步与复制问题。这些内容我们将在第 5 章介绍，提供一些可参考的通用思路。

二、挑战性能瓶颈

从第 5 章开始将介绍网站 PV 量从 1 亿到 50 亿的发展过程中应用系统遇到的各种性能瓶颈，以及我们的解决方案。这个过程可大致分为 3 个时间段，如图 0.3 所示。

试读结束，需要全本PDF请购买 www.ertongbook.com

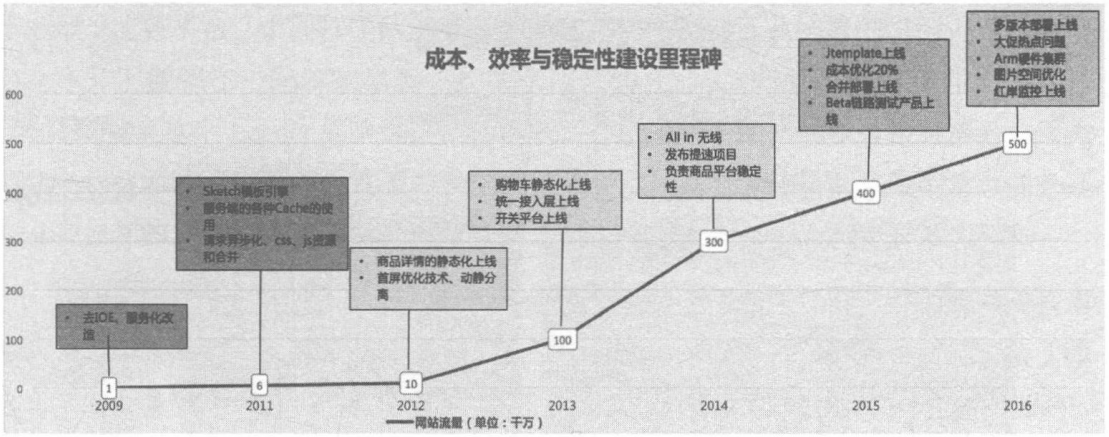


图 0.3　网站 PV 量从 1 亿发展到 50 亿的过程

第 1 个时间段：主要解决网站的易用性和扩展性问题。例如当时提的“去 IOE”就是将数据库从 Oracle 迁到 MySQL 上，通过分库分表来解决扩展性问题，同时做了很多网页端的优化工作，如 JavaScript（以下简称 JS）的异步加载、首屏优先渲染等。

第 2 个时间段：这段时间网站的流量增长非常迅速，每年双 11 的流量更是异常疯狂，系统出现了性能瓶颈。记得某次评估双 11 的流量，光详情页系统就需要增加上万台服务器，简直无法想象！所以必须考虑用非常规手段来优化性能。这个阶段我们主要通过静态化技术来解决读系统的性能瓶颈，大概用了 1~2 年的时间来持续迭代，直到实现了静态化系统改造才算彻底解决了性能问题，使系统能够支持上百万的 QPS。

第 3 个时间段：由于业务系统的复杂性上升非常快、业务的耦合度比较高，给系统的稳定性带来很大挑战，所以这个阶段进行了系统的稳定性建设，产出了很多稳定性工具和产品，另外性能优化也更朝着架构优化的方向发展。

我们还做了很多提升应用性能和开发效率的工作，从成本的角度来看，可以把这些工作归结为如图 0.4 所示的内容。

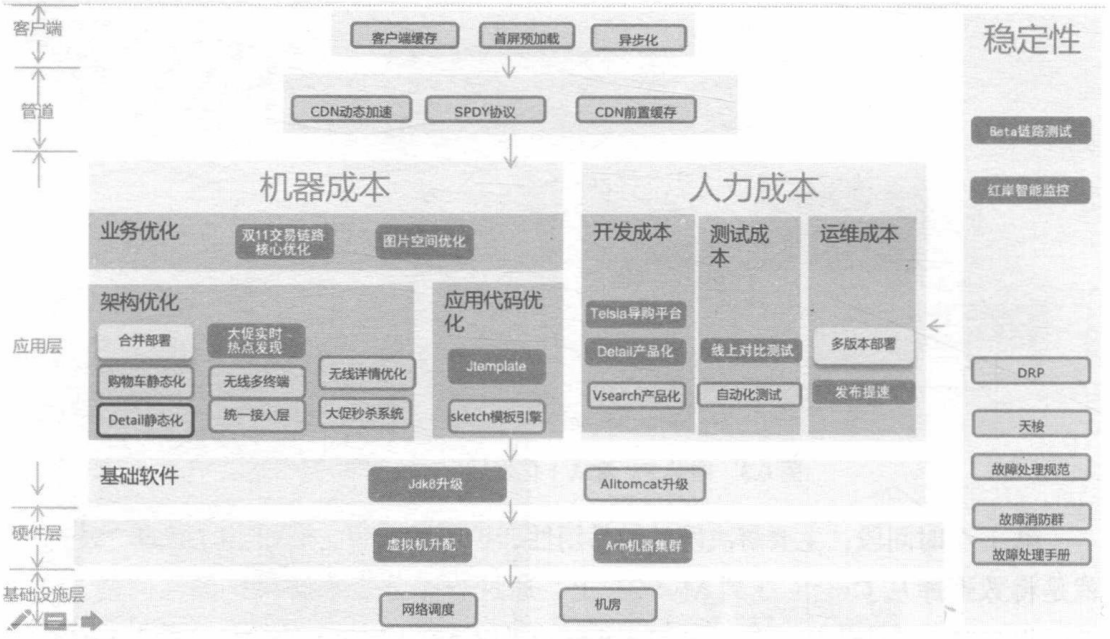


图 0.4　主要工作的归类

从端到中间的管道、再到应用层以及基础设施的优化，本书会分成 4 项内容来介绍：应用层代码优化、应用架构的优化、全链路的优化和基础设施的优化。

最后一章我们会介绍在整个网站飞速发展的同时，如何在进行架构的升级、应对突发的大流量这些极端的情况下，仍能保持整体网站的高可用和稳定性。至于如何做好双 11 的稳定性，这些内容已有多次的分享，本章会做一些系统的介绍。

三、组织架构的变迁

如果说技术是生产力，那么组织就决定了生产关系，而生产关系要适应生产力的发展。笔者当时所在的团队是业务平台，可以理解成技术大团队中的架构师团队，本身没有具体的业务产品，专注于解决一些横向的、架构上的问题。一般每个重要项目都会有一名架构师参与，业务平台和业务的关联比较直接、紧密。

中间件团队和业务平台相比，则和业务的距离稍微远一点，专注于业务开发过程中偏公共和通用的技术组件。这个团队和业务也相对比较紧密，业务开发中一些通用的技术组件，例如同步远程调用 RPC 框架、异步通信消息框架、业务动态配置框架、Web 开发框架、分布式数据层、分布式 Session 框架等，都是在业务开发中会用到且需要统一和规范使用的通用组件。

由于业务平台、中间件团队和业务的关系相对紧密，因此团队的组织关系一定要和业务开发捆绑在一起。在早期，这两个团队的 Leader 最好也是业务开发团队的 Leader，而且这个 Leader 要有强力的话语权。否则这两个团队的业务很难落地。架构师和中间件团队可以理解为业务开发中偏精英一点的团队，从长期发展来看，业务开发也有强烈的意愿要往偏技术的方向发展，所以他们的关系不一定是长期稳定的。

当技术团队人员扩张到接近 1000 人时，业务平台就慢慢解散了：因为随着技术团队分工越来越细，它的专业度也会越来越深，统一的架构师团队显然已经无法支撑这么多的业务团队了。在这种情况下，业务平台就被拆解分散到各个技术产品团队中，业务平台的 Leader 也成为新的技术团队的 Leader。该演进过程如图 0.5 所示。

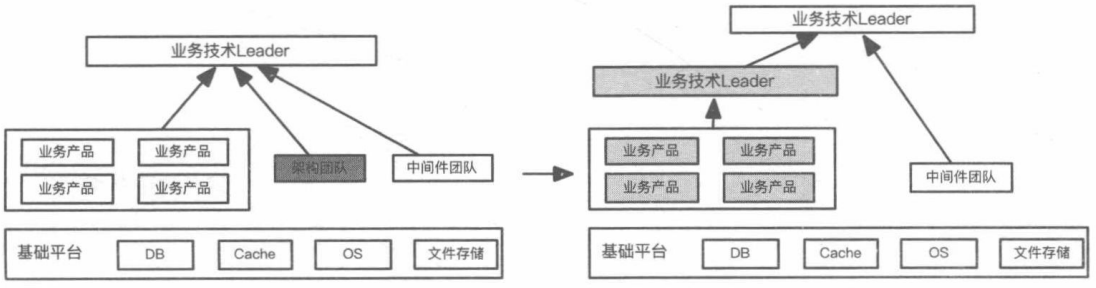


图 0.5　组织架构的可能演进

在图 0.5 中，中间件团队在建立伊始，一定要和业务团队保持紧密关联，同时，一些中间件产品最好是发展得比较完善后，再独立成图 0.5 右侧的结构，这会比较好。如果从一开始，中间件团队就保持了图 0.5 右侧的图的结构，在落地时会比较困难。

不过凡事都不是绝对的，人与人之间最大的问题其实就是信任问题。公司内部同事之间的协作最大的问题来自于利益的分配，团队之间合作最和谐的方式就是既有合作又有制约。这其中很多都是出于业务上的考虑。除了上面所介绍的，还有一个在组织结构上对技术影响比较大的事件——中台的提出——我们后面会介绍。

四、工程师文化的形成

在我看来，所谓的工程师文化是指就算没有 KPI 的考核，大家仍然还会认为某些事情是应该做的，而某些事情是不应该做的。例如，遇到问题必须追根溯源、敬畏线上的每次变更并力保稳定性、在技术沟通上对事不对人、简单坦诚、做事不给自己也不给别人挖坑等。

笔者感受比较深的是如果一家公司有一个类似双 11 这样的集中活动，那么对公司就是一次技术水平的大考，在这种压力下，在技术上要追求极致，工程师文化就自然而然地、有默契地形成了。因此，像双 11 这种能取得大家共识的公共事件可以在很大程度上推动技术的发展。

工程师文化和每个公司目前所处的环境和阶段也是息息相关的。如果一个创业公司每天面临着如何生存的问题，而员工却在考虑技术的设计是否完美，显然不是太现实。在这种情况下，业务开发的效率和稳定性会更重要。不过，长久来看，该还的债终究是要还的，一个好的工程师文化的建立对公司未来的良好发展至关重要。

目　录

试读结束，需要全本PDF请购买 www.ertongbook.com