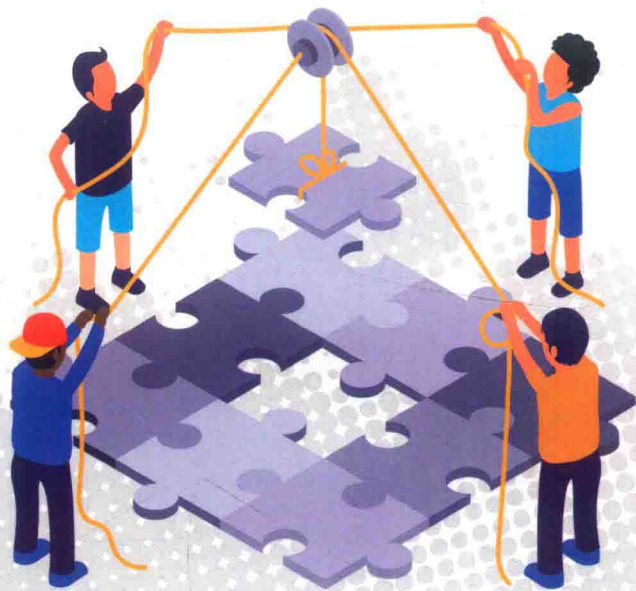


# 小团队构建大网站

## 中小研发团队架构实践

张辉清 / 等著



非外借

# 小团队构建大网站 中小研发团队架构实践

张辉清 杨丽 / 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书结合作者近几年的工作经验，总结了一套可直接落地、基于开源、成本低、可快速搭建的中小研发团队架构实践方法。本书共 5 篇 22 章，开篇是本书的导读；架构篇是设计思想的提升，包括企业总体架构、应用架构设计、统一应用分层等；框架篇主讲中间件和工具的使用，包括消息队列、缓存、Job、集中式日志、应用监控和微服务等；公共应用篇是技术与业务的结合，包括单点登录和企业支付网关；进阶篇是从架构到管理，包括技改案例、技术与业务的匹配与融合等。从架构、框架、公共应用，到案例实战和技术管理，本书将大公司的工程理念压缩应用到中小研发团队，使小团队也能构建大网站。

本书不仅适用于高级程序员、架构师、CTO，也适用于 IT 项目经理、技术经理，以及对架构技术感兴趣的中高级软件开发从业者。

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

小团队构建大网站：中小研发团队架构实践 / 张辉清，杨丽著. —北京：电子工业出版社，2019.1

ISBN 978-7-121-35215-7

I. ①小… II. ①张… ②杨… III. ①网站建设 IV. ①TP393.092.1

中国版本图书馆 CIP 数据核字 (2018) 第 238878 号

责任编辑：陈晓猛

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱

开 本：720×1000 1/16

版 次：2019 年 1 月第 1 版

印 次：2019 年 1 月第 1 次印刷

定 价：69.00 元

邮编：100036

印张：14.5



千字

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 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

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

# 前言

**需求：**没有人会读前言？！

**解决方案：**把前言内容写好，并作为开篇第 1 章。



用户画像

## 读者服务

轻松注册成为博文视点社区用户（[www.broadview.com.cn](http://www.broadview.com.cn)），扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交。若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/35215>





# 目录

第 1 篇 开篇.....	1
1 可参考的才是有价值的（含案例和代码）.....	2
1.1 框架篇——工欲善其事，必先利其器.....	2
1.2 架构篇——思想提升.....	7
1.3 公共应用篇——业务与技术的结合.....	8
1.4 进阶篇——从架构到管理.....	9
1.5 案例参考和 Demo 下载.....	10
第 2 篇 架构篇.....	11
2 企业总体架构.....	12
2.1 企业商务模型.....	13
2.2 架构现状.....	13
2.3 领域模型.....	16
2.4 架构规划.....	17
2.5 架构实施.....	24
2.6 案例参考.....	25
3 应用架构设计.....	26
3.1 初识架构设计.....	26

3.2	应用架构设计案例 .....	27
3.3	更多知识探讨 .....	32
3.4	互联网公司的架构设计要怎么落地.....	38
3.5	你给技术打个分 .....	40
3.6	案例参考 .....	40
<b>4</b>	<b>统一应用分层 .....</b>	<b>41</b>
4.1	为什么要统一应用分层 .....	41
4.2	统一应用逻辑架构 .....	42
4.3	分层规范实践 .....	44
4.4	互动问答 .....	50
4.5	Demo 下载.....	51
<b>5</b>	<b>生产环境诊断工具 WinDbg.....</b>	<b>52</b>
5.1	诊断工具简介 .....	52
5.2	获取异常进程的 Dump 文件 .....	53
5.3	WinDbg 的使用方法.....	57
5.4	一个真实案例 .....	60
5.5	Demo 下载.....	62
<b>第 3 篇</b>	<b>框架篇 .....</b>	<b>63</b>
<b>6</b>	<b>RabbitMQ 快速入门及应用 .....</b>	<b>64</b>
6.1	为什么要用消息队列 RabbitMQ .....	64
6.2	RabbitMQ 简介 .....	66
6.3	RabbitMQ 的工作原理 .....	67
6.4	RabbitMQ 的基本用法 .....	67
6.5	Demo 下载.....	68
<b>7</b>	<b>Redis 快速入门及应用 .....</b>	<b>69</b>
7.1	Redis 简介 .....	69
7.2	Redis 的数据结构 .....	70

7.3	Redis 的重要特性 .....	71
7.4	使用方法 .....	72
7.5	Redis Key 命名规范与常见问题 .....	73
7.6	Demo 下载 .....	74
8	任务调度 Job .....	75
8.1	Job 简介 .....	75
8.2	WinJob .....	75
8.3	HttpJob .....	77
8.4	Cron 表达式 .....	79
8.5	Demo 下载 .....	81
9	应用监控系统 Metrics .....	82
9.1	Metrics 简介 .....	82
9.2	埋点 Metrics.NET 的方法 .....	82
9.3	Grafana 配置 .....	84
9.3.1	设置仪表盘 (Dashboard) .....	85
9.3.2	设置面板 (Panel) .....	86
9.3.3	设置模板 Templating .....	89
9.3.4	设置 Time Range .....	91
9.3.5	告警设置 .....	92
9.4	其他说明 .....	94
9.5	Metrics 的使用价值 .....	94
9.6	Demo 下载 .....	95
10	集中式日志 ELK .....	96
10.1	集中式日志 .....	96
10.2	配置方法 .....	97
10.3	使用方法 .....	100
10.4	Demo 下载 .....	102
11	微服务架构 MSA .....	103
11.1	MSA 简介 .....	103



11.2	MSA 框架的使用 .....	105
11.3	微服务治理 .....	109
11.4	微服务网关 API Gateway .....	110
11.5	Demo 下载.....	113
<b>12</b>	<b>搜索服务 Solr.....</b>	<b>114</b>
12.1	Solr 简介.....	114
12.2	Solr 的工作原理.....	115
12.3	Solr 的特性.....	122
12.4	Demo 下载.....	123
<b>13</b>	<b>分布式协调器 ZooKeeper .....</b>	<b>124</b>
13.1	ZooKeeper 是什么 .....	124
13.2	ZooKeeper 的工作原理简介 .....	124
13.3	ZooKeeper 的典型应用场景 .....	126
13.4	Demo 下载.....	128
<b>14</b>	<b>小工具合集.....</b>	<b>129</b>
14.1	ORM 工具 .....	129
14.2	对象映射工具 .....	131
14.3	IoC 工具 .....	134
14.4	DLL 包管理工具.....	135
14.5	Demo 下载.....	141
<b>15</b>	<b>一键发布和测试之持续集成工具 Jenkins.....</b>	<b>142</b>
15.1	Jenkins 简介 .....	142
15.2	Jenkins 插件与相关工具 .....	143
15.3	Jenkins 关键配置 .....	144
15.4	Jenkins 的使用价值 .....	151

第 4 篇 公共应用篇 .....	153
16 单点登录 .....	154
16.1 单点登录简介 .....	154
16.2 SSO 技术实现 .....	155
16.3 JWT 规范 .....	159
17 企业支付网关 .....	160
17.1 企业支付网关介绍 .....	160
17.2 统一支付服务 .....	161
17.3 统一支付通知 .....	164
17.4 Demo 下载 .....	165
第 5 篇 进阶篇 .....	167
18 技改之路：从单体应用到微服务 .....	168
18.1 系统背景 .....	168
18.2 前期工作 .....	170
18.3 技改实施 .....	172
18.4 总结 .....	177
18.5 互动问答 .....	178
19 机票垂直搜索引擎之性能优化 .....	180
19.1 行业背景与垂直搜索 .....	180
19.2 主要问题与解决方案 .....	181
19.3 静态数据与任务打底 .....	181
19.4 缓存策略与数据一致 .....	183
19.5 实时查询与三段超时 .....	184
19.6 政策匹配与算法优化 .....	185
19.7 小结 .....	188

20	上云纪要 .....	189
20.1	为什么要上云 .....	189
20.2	内部虚拟化和外部云化 .....	190
20.3	云选型 .....	191
20.4	上云八条 .....	192
20.5	成功上云 .....	193
20.6	上云总结 .....	194
21	技术与业务的匹配与融合 .....	196
21.1	技术人员与业务人员的抱怨 .....	196
21.2	问题出在哪里 .....	197
21.3	理解源于彼此的了解 .....	198
21.4	如何去匹配与融合 .....	199
21.5	什么在驱动公司的发展 .....	201
22	研发团队文化是怎么“长”出来的 .....	202
22.1	神秘的文化 .....	202
22.2	遇到的问题 .....	203
22.3	解决之道 .....	203
22.4	总结与提升 .....	207
22.5	“长”出来的团队文化 .....	207
	后记 .....	209
	架构师进阶之路 .....	211
	谈谈互联网公司的技术架构和管理 .....	213
	短评 .....	216

第 1 篇

开 篇

# 1

## 可参考的才是有价值的 (含案例和代码)

技术大会上的分享大多“高大上”——亿级流量、超大型研发团队，虽然值得借鉴，但由于应用场景与研发资源的差异，一般企业并不容易落地。其实，中小型研发团队在 IT 行业还是占大多数，他们在技术架构方面的问题较多，技术阻碍业务、跟不上业务发展的情况很常见。笔者是一个有十多年经验的 IT 老兵，曾在两家几千人的技术团队做过架构与技术管理工作，也曾在几十人至几百人的中小研发团队做过首席架构师和 CTO。在互联网大厂做技术研发工作，大多数人只是一个“螺丝钉”。而在中小研发团队做研发工作，则比较容易掌控全局。本书结合笔者近几年的工作经验，摸索出一套可直接落地、基于开源、成本低、可快速搭建的框架及架构方案。本书贴近一般程序员的实际情况，更具应用和参考的价值，共 5 篇 22 章，包括开篇、框架篇、架构篇、公共应用篇和进阶篇，以下是具体介绍。

### 1.1 框架篇——工欲善其事，必先利其器

如果说运维是地基，那么框架就是承重墙。盖房子是先打地基，再建承重墙，最后才垒砖，所以中间件的搭建和引进是建设高可用、高性能、易扩展、可伸缩的大中型系统的前提。框架篇中的每一章主要由四部分组成：基本概念、工作原理、使用场景及可

直接调试的 Demo。其中中间件及 Demo 历经两家公司四年时间的考验,涉及几百个应用、100 多个库和 1 万多张表,日订单从几万到十几万,年 GMV 从几十亿到几百亿。所有中间件与工具都基于开源。早期也有部分中间件和工具是自主研发的,比如集中式日志和度量框架,后期在第二家公司时为了快速搭建、降低成本、易于维护和扩展,全部改为开源软件。这样不仅利于个人的学习成长、知识重用和职业生涯发展,也利于团队的组建和人才的引进。

### 1. 集中式缓存 Redis

缓存是计算机的难题之一,分布式缓存亦是如此。Redis 看起来非常简单,但它影响系统的效率、性能和数据一致性。用好它不容易,包括缓存时长(复杂多维度的计算)、缓存失效处理(主动更新)、缓存键(Hash 和方便人工干预)、缓存内容及数据结构的选择、缓存雪崩的处理、缓存穿透的处理等。Redis 除了缓存的功能,还有其他功能,比如 Lua 计算能力、Limit 与 Session 时间窗口、分布式锁等。我们使用 ServiceStack.Redis 做客户端,使用方法详见 Demo。

### 2. 消息队列 RabbitMQ

消息队列好比葛洲坝,有大量数据的堆积能力,再可靠地进行异步输出。它是 EDA 事件驱动架构的核心,也是 CQRS 同步数据的关键。为什么选择 RabbitMQ 而没有选择 Kafka? 是因为业务系统对消息有高可靠性要求,以及对复杂功能(如消息确认)的要求。

### 3. 集中式日志 ELK

日志主要分为系统日志和应用日志两类。试想一下,如何在一个具有几百台服务器的集群中定位问题?如何追踪每天产生的几 GB 甚至几 TB 的数据?集中式日志就是此类问题的解决方案。早期我们使用自主研发的 Log4Net+MongoDB 来收集和检索日志信息,但随着数据量的增加,查询速度却变得越来越慢。后期改为开源的 ELK,虽然易用性有所下降,但它支持海量数据及与编程语言无关的特征。

### 4. 任务调度 Job

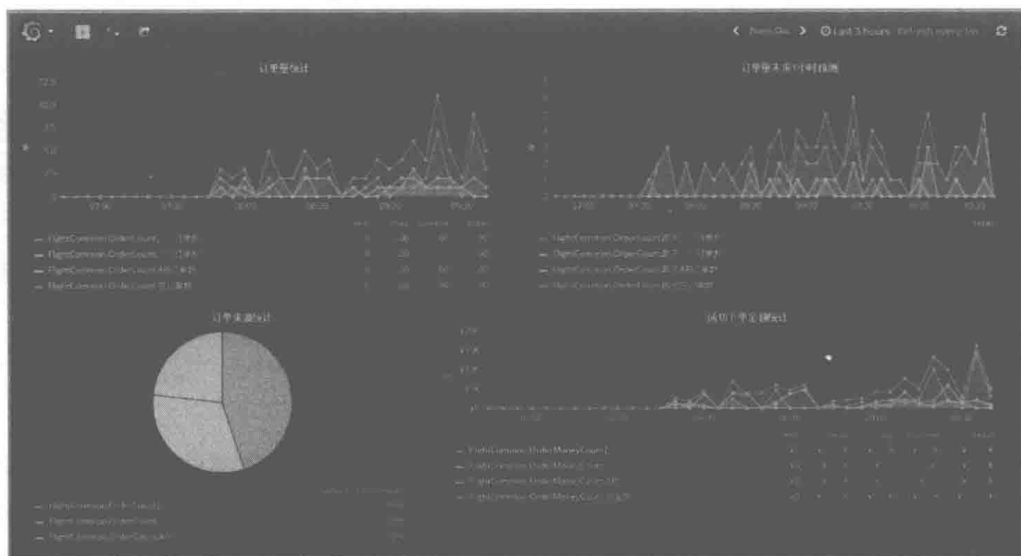
任务调度 Job 如同数据库作业或 Windows 计划任务,是分布式系统中异步和批处理的关键。我们的 Job 分为 WinJob 和 HttpJob: WinJob 是操作系统级别的定时任务,使用开源的框架 Quartz.NET 实现;而 HttpJob 则是自主研发实现的,采用 URL 方式可定时调

用微服务。HttpJob 借助集群巧妙地解决了 WinJob 的单点和发布问题，并集中管理所有的调度规则，调度规则有简单规则和 Cron 表达式。HttpJob 简单易用，但间隔时间不能低于 1 分钟，毕竟通过 URL 方式来调度并不高效。下图是 HttpJob 的管理后台。



## 5. 应用监控 Metrics

“没有度量就没有提升”，度量是改进优化的基础，是做好一个系统的前置条件。Zabbix 一般用于系统级别的监控，Metrics 则用于业务应用级别的监控。业务应用是个“黑盒子”，通过数据埋点来收集应用的实时状态，然后展示在大屏或看板上。它是报警系统和数字化管理的基础，还可以结合集中式日志来快速定位和查找问题。我们的业务监控系统使用 Metrics.NET+InfluxDB+Grafana，如下图所示。



## 6. 微服务框架 MSA

微服务是细粒度业务行为的重用，需要与业务能力及业务阶段相匹配。微服务框架是实现微服务及分布式架构的关键组件，我们的微服务框架是基于开源 ServiceStack 实现的。它简单易用、性能好，文档自动生成、方便调试测试，调试工具是 Swagger UI，自动化接口测试工具是 SoapUI。微服务的接口开放采用自主研发的微服务网关，配置治理后台即可。网关以 NIO、IOCP 的方式实现高并发，主要功能有鉴权、超时、限流、熔断、监控等。下图是 Swagger UI 调试工具。

POST /OrderService/Create PNR 文本下单

Response Class (Status)  
Model | Model Schema

Create **Response** {  
Code (int): 返回编码,  
Message (string, optional): 返回消息,  
ContactInfo (ContactInfo, optional): 联系人信息,  
OrderNo (string, optional): 订单号,  
OrderState (int): 订单状态,  
Passengers (Array[OrderPassenger], optional): 旅客列表,  
PayPrice (double): 订单总金额,  
PNR (string, optional): PNR,  
PNRText (string, optional): PNR内容,  
PolicyID (string, optional): 政策ID,  
Segments (Array[OrderSegment], optional): 航程列表  
}

Response Content Type **application/json**

Parameters

Parameter	Value	Description	Parameter Type	Data Type
	<input type="text"/>		form	string
PNRText	<input type="text"/>	PNR内容	form	string

body

```
{  
  "ContactInfo": {  
    "Contact": "",  
    "Tel": ""  
  },  
  "Passengers": [  
    {  
      "Birthday": "",  
      "CardNo": "",  
      "CardType": ""  
    }  
  ]  
}
```

Parameter content type: **application/json**

body

Model | Model Schema

```
{  
  "ContactInfo": {  
    "Contact": "",  
    "Tel": ""  
  },  
  ...  
}
```

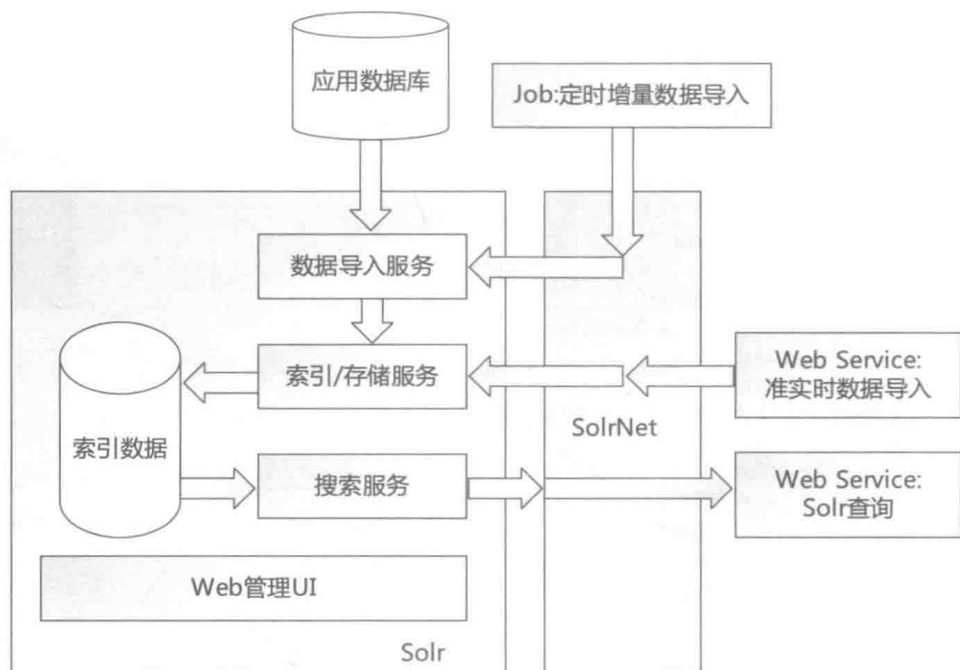
Click to set as parameter value

Try it out!



## 7. 搜索服务 Solr

分库分表后的关联查询，大段文本的模糊查询，这些要如何实现呢？显然传统的数据库没有很好的解决办法，这时可以借助专业的检索工具。全文检索工具 Solr 不仅简单、易用、性能好，而且支持海量数据高并发，只需实现系统两边数据的准实时或定时同步即可。下图是 Solr 的工作原理。



## 8. 更多工具

- 分布式协调器 ZooKeeper：工作原理、配置中心、Master 选举、Demo。
- ORM 框架：Dapper.NET 语法简单、运行速度快，与数据库无关，SQL 自主编写可控，是一款适合互联网系统的数据库访问工具。
- 对象映射工具 EmitMapper 和 AutoMapper：EmitMapper 的性能较高，AutoMapper 的易用性较好。
- IoC 框架：控制反转（IoC）轻量级框架 Autofac。
- DLL 包管理：公司内部 DLL 包管理工具 NuGet，可解决 DLL 集中存储、更新、引用、依赖的问题。