摘 要

随着作业帮业务线的不断壮大与发展,作为其中直接面对来自客户请求的服务端，无疑是要同时承受更多的请求，并为用户提供更好的体验。这时服务端的性能常常会成为业务发展的瓶颈。在工作场景中面对持续变化的流量模式、不均衡的基础设施特征、以及持续变化的瓶颈，要对需求方提出的功能进行开发并维持各个功能模块的稳定性，保证各个接口请求处理速度与增加并发请求的最大数量，便是工作中的重点与挑战。

**关键词：**服务端，高并发，稳定性，请求

Abstract

With the continuous development of Zuoyebang’s business.,As a direct face from the customer request, the server is undoubtedly to bear more requests at the same time, and provide users with a better experience. At this time the performance of the server side will often become the bottleneck of business development. In the work scene ,in the face of continuous changes in the flow patterns, unbalanced infrastructure characteristics, as well as the continuous change of application’s bottleneck, according to the requirements of the demand side to to develop and maintain the stability of the various functional modules to ensure that each interface requests processing speed and Increasing the maximum number of concurrent requests is the focus and challenge of the job.

**Keywords:**server, High-Concurrency, stability,request

目 录

[第一章 毕业设计（顶岗实习）概况 1](#_Toc451356829)

[1.1 实习单位与实习岗位概况，对企业所属行业的认识 1](#_Toc451356830)

[1.2 实习项目课题背景、价值、意义以及国内外研究现状 1](#_Toc451356831)

[1.3 实习项目整体执行完成情况概述 1](#_Toc451356832)

[第二章 复杂工程问题归纳与实施方案可行性研究 3](#_Toc451356833)

[2.1 ×××××× 3](#_Toc451356834)

[2.1.1 ×××××× 3](#_Toc451356835)

[2.1.2 ×××××× 3](#_Toc451356836)

[2.1.3 ×××××× 3](#_Toc451356837)

[2.2 ×××××× 3](#_Toc451356838)

[2.3 ×××××× 3](#_Toc451356839)

[第三章 针对复杂工程问题的方案设计与实现 5](#_Toc451356840)

[3.1 ×××××× 5](#_Toc451356841)

[3.1.1 ×××××× 5](#_Toc451356842)

[3.1.2 ×××××× 5](#_Toc451356843)

[3.1.3 ×××××× 5](#_Toc451356844)

[3.2 ×××××× 5](#_Toc451356845)

[3.3 ×××××× 5](#_Toc451356846)

[第四章 知识技能学习情况 7](#_Toc451356847)

[4.1 ×××××× 7](#_Toc451356848)

[4.1.1 ×××××× 7](#_Toc451356849)

[4.1.2 ×××××× 7](#_Toc451356850)

[4.1.3 ×××××× 7](#_Toc451356851)

[4.2 ×××××× 7](#_Toc451356852)

[4.3 ×××××× 7](#_Toc451356853)

[第五章 工程计划管控与执行情况 9](#_Toc451356854)

[5.1 ×××××× 9](#_Toc451356855)

[5.1.1 ×××××× 9](#_Toc451356856)

[5.1.2 ×××××× 9](#_Toc451356857)

[5.1.3 ×××××× 9](#_Toc451356858)

[5.2 ×××××× 9](#_Toc451356859)

[5.3 ×××××× 9](#_Toc451356860)

[第六章 职业素养与工程伦理的学习与培养 11](#_Toc451356861)

[6.1 ×××××× 11](#_Toc451356862)

[6.1.1 ×××××× 11](#_Toc451356863)

[6.1.2 ×××××× 11](#_Toc451356864)

[6.1.3 ×××××× 11](#_Toc451356865)

[6.2 ×××××× 11](#_Toc451356866)

[6.3 ×××××× 11](#_Toc451356867)

[第七章 对软件工程实践以及软件工程领域发展的认识 13](#_Toc451356868)

[7.1 ×××××× 13](#_Toc451356869)

[7.1.1 ×××××× 13](#_Toc451356870)

[7.1.2 ×××××× 13](#_Toc451356871)

[7.1.3 ×××××× 13](#_Toc451356872)

[7.2 ×××××× 13](#_Toc451356873)

[7.3 ×××××× 13](#_Toc451356874)

[第八章 结束语 15](#_Toc451356875)

[8.1 本文内容 15](#_Toc451356876)

[8.2 顶岗实习项目课题有待进一步解决的问题及方向 15](#_Toc451356877)

[8.3 对软件工程实践以及软件工程领域发展的认识 15](#_Toc451356878)

[8.4 本人毕业设计（顶岗实习）收获及体会 15](#_Toc451356879)

[参考文献 17](#_Toc451356880)

[致谢 19](#_Toc451356881)

[外文资料原文 21](#_Toc451356882)

[外文资料译文 23](#_Toc451356883)

第一章 毕业设计（顶岗实习）概况

1.1 实习单位与实习岗位概况，对企业所属行业的认识

作业帮教育科技（北京）有限公司是百度旗下的教育科技公司,原本隶属于百度知道的架构体系下。2013年12月作业帮官方内测qq群在百度内部成立，经过两年的发展，作业帮迅速占领市场。作为教育市场最有竞争力的app，2015年5月，教育部召开信息化教学调研会上，百度“作业帮”成为示范案例。同年作业帮用户数突破6000万。为了作业帮更好的发展，百度知识搜索体系负责人侯建斌向李彦宏申请将作业帮独立运营。经过一段时间的准备，的2015年9月2日，作业帮作为百度“航母计划”的一部分宣布分拆， A轮由红杉资本、君联资本联合投资. 2015年10月31日，作业帮教育科技（北京）有限公司正式独立运营。公司16年又获得6000万美元b轮融资。作业帮主体产品作业帮APP是面向全国中小学生的移动学习平台，也是习题搜索、高效练习和学习沟通的综合学习工具。如今已有拍照搜题、同步练习、老师答疑、语音搜题、作文搜索、求助学霸、在线直播课等多项业务功能。

公司所在的互联网教育行业是当前缓解或解决教育资源分配不均的最佳途径，国家对互联网教育继续给予大力政策扶持。2014年11月，教育部等五部门联合下发《构建利用信息化手段扩大优质教育资源覆盖面有效机制的实施方案》。2015 年 4 月，教育部出台《关于加强高等学校在线开放课程建设应用与管理的意见》，为慕课发展营造良好的政策环境。2015年7月，《国务院关于积极推进“互联网+”行动的指导意见》提出加快发展基于互联网的教育新兴服务，探索新型教育服务供给方式。这些政策的出台，为互联网教育营造了良好发展环境。

我实习所在作业帮平台服务端研发岗是为作业帮整个教育平台提供基础架构服务的重要部门。部门负责了公司后端服务的的基础框架搭建，完成了公司的用户系统、消息系统、广告系统、检索系统、支付系统等核心模块的开发。同时负责公司核心产品作业帮app版本迭代过程中的服务端功能开发，负责分析产品方提出的需求，然后对相关功能进行技术调研、架构设计并整理出相关方案文档。考虑功能的应用场景、并发量等因素进行服务端代码编写，接口设计、数据库设计、缓存设计、集群服务调度。并在完成功能代码后，进行联调，功能测试、回归、上线。并负责上线之后的数据回归等工作。

1.2 实习项目课题背景、价值、意义以及国内外研究现状

在2016年，公司经历了飞速的成长， 整体团队人员扩张了数倍， 公司整体业务线从单一的拍照搜提扩张到支持多个多样化在线教育模块。如今作业帮作为互联教育领域的领军者，日活已达千万级别，瞬时压力会达数万qps。检索数据表每日新增数千万行，在这种业务场景下，同时又是资源有限的环境下必须思考如何靠小规模集群承载大量请求。如今国内外大规模互联网公司都面临着高瞬时并发的挑战。国内顶级服务端设计如淘宝在高峰期间都有着单个上百万qps的压力。国外如谷歌、推特等公司靠着几十万上百万的服务器集群承载着世界各地发送来的请求。虽然面对高并发场景构建稳定的服务端的业务场景已经有很多的公司已经完成的很好，大公司的特定业务线都有着较为成熟的模型去应对所谓的高并发场景，但由于每个公司的业务场景的差异性，再优秀的架构都不具备通用性，需要根据每种应用特点针对性来设计，且不同公司对于高并发的核心技术都并非透明，故在集群资源的有限性等因素下，针对所在的公司特定化设计出特定的架构维持业务的可靠性、保证足够低的成本是非常具有挑战性的。

1.3 实习项目整体执行完成情况概述

在实习期间主要的项目有黑板报的四次改版开发，作业帮账号系统重建及同步的第三方登录功能开发，作业帮微课功能开发，广告审核系统开发，及一些如作业帮三周年活动、开学减负等运营活动的服务端功能开发。

在黑板报的多次改版过程中，由最初的简单的文章列表功能进行多次功能的改进和优化。最初的第一个改版实现了文章详情页的点赞，弹幕功能，第二次改版实现了获取腾讯天天快报的第三方内容作为独立的栏目展示，丰富了黑板报列表并实现列表的多种样式——多图、组图、图文的混排列表形式。第三版实现了作业帮原创内容与腾讯等第三方内容的混排功能，极大的增加了用户的停留市场。并在第四次改版增加发布短视频内容，并新增了百度秒懂百科的数据源再一次丰富了黑板报模块功能，加强了用户体验。并在几次改版过程中，还在后台添加了文章定时上线下线等功能。技术层面上在热点资源处都加上了一层缓存，极大的减少了数据库的压力，并随着几次改版的，将内容的存储数据结构重构，加强了数据格式的可拓展性。在不断迭代开发中，黑板报的点击、停留时长等数据都较之前有了巨大的提升。

关于作业帮的账号系统重建则是实习期间经手的最大规模项目，直接影响了作业帮整个用户体系的过亿用户。作业帮最初所有用户都被整合成为手机注册，故作业帮核心用户表为两个表，表a是以手机为主键的存储用户session，uid,password,表b是以uid作为主键存储用户信息的表。在增加了第三方登录后，用户的session，password就并非和手机有强关联关系。所以需要将session、password等字段拆为单独的表进行存储。需要进行的进行表数据迁移工作涉及全用户系统的上亿用户，涉及的接口包含登陆、注册、修改密码、单点登录、忘记密码等功能。所有的接口不能影响线上业务线，故在接口不变更的情况下重构底层服务端逻辑。重构完成后进行线下测试、内网测试、小流量测试、一致性测试后才能开启全量。项目前后经历较长时间，并于三月底上线，线上0用户反馈前后出现因账号系统迁移导致了账号不一致、数据错误等问题，线上回归正常，成功实现了账号迁移。

作为一个教育类app，为用户提供优质的体验同时谋求良好的变现之道是必要的。作为互联网公司通用的变现途径之一就是通过广告。接入第三方广告平台的广告，请求第三方所提供的接口就会返回相应的广告资源通过站展现点击，就可从中获得收益。但由于广告平台投放的广告质量并非可控，作为教育平台不适宜出现的广告种类非常多。故在接入广告平台的基础上拓展作业帮现有的广告平台来源并在与广告平台对接中新增一层审核接口，已保存投放广告的优质性。审核功能包括广告审核、广告主审核。所有未被审核过的内容被请求过后不会被直接展现而是进入后台页面进行人工审核。该功能重构了广告请求接口，并新增了相应的后台服务。上线过后，广告的质量得到了把控，并通过将审核结果缓存，在添加了审核之后也并未给服务端增加过多压力。

作业帮微课功能开发是基于为高年级学生提供优质定时教育服务的初衷进行开发的。模块主要包括两个功能——学霸都在看和学霸都在练。

第二章 复杂工程问题归纳与实施方案可行性研究

2.1 黑板报改版问题

2.1.1 增加热点数据缓存与增加点赞弹幕功能

作为依托于大流量app作业帮的黑板报，它天然就有着较大的潜在流量。在黑板报上线了一段时间后，作业帮的日均请求量已达到一定规模。为了降低db压力，并降低用户的请求时长提升用户体验，决定在列表页和热点的文章的详情页增加缓存。

文章详情页的缓存较为简单，缓存系统可直接选择memcached，将文章id作为key，文章详情作为value进行缓存。但同时涉及的是，黑板报进行增删改操作不仅要对db进行操作还需对cache进行相应的操作。

较详情页缓存较为复杂的是列表缓存，作为用户阅读黑板报必须进入的列表页面，它采用feed流样式。简单来说就是所有的数据请求都是异步，请求类型分为三种，首次进入的首屏数据请求，上拉更新操作与下拉加载操作。最初由数据库实现的时候特别简单——下拉时根据上线时间排序查找表中上线时间小于请求刷新时前端传来的时间戳，下拉时根据上线时间排序查找表中上线时间大于请求刷新时前端传来的时间戳。但增加缓存时需要考虑的则居多，首先缓存的目标对象应该是热点对象。所以我将缓存对象设为前十屏的内容。缓存将文章列表id数组作为单独对象处理，再将前十屏列表每一个文章对象单独缓存。在缓存有效期内请求缓存的id列表，读到应该加载的文章的id，再根据id得到缓存的文章对象拼凑成json对象返回给前端进行展示。

同时，使用缓存时应注意避免单个节点压力过大以致瘫痪。故应注意平衡缓存集群之间的压力。因为列表页为最热数据，且只有单key。则必然存储该key的节点会承受更高的压力。所以将相同id列表做多个数据在多个节点存储，每次请求随机取其中一个key，让请求分散到不同节点，均衡各个节点之间的压力。

同时鉴于单纯的文章列表展示太过普通缺乏用户之间的交流且粘性较低。从用户角度思考，产品方提出了在黑板报详情页增加k12用户喜欢的弹幕功能和点赞功能（弹幕功能和点赞功能见图2-1）。弹幕数据作为独立的数据则采用新建表去存储，以文章为主键存储弹幕内容用户id等内容。点赞数据则直接存储在黑板报的详情表中。因为弹幕数据的访问量和文章内容一样属于热点数据故也需要加上一层缓存。



图2-1 点赞与弹幕功能

2.1.2 增加腾讯天天快报数据源与优化数据库存储

……

2.1.3 ××××××

……

2.2 ××××××

……

2.3 ××××××

……

第三章 针对复杂工程问题的方案设计与实现

3.1 ××××××

3.1.1 ××××××

……

3.1.2 ××××××

……

3.1.3 ××××××

……

3.2 ××××××

……

3.3 ××××××

……

这是论文全文的重点，主要包括：

1. 在详细需求分析的基础上，对复杂软件工程问题进行推理、分析，详尽阐述针对复杂软件工程问题而开展的总体设计和详细设计。
2. 在详细设计的基础上，进一步阐述针对复杂工程问题的具体实现、测试（调试）以及结果分析。

第四章 知识技能学习情况

4.1 ODP&RAL&NMQ

4.1.1 ODP

ODP（见图3-1）是百度发布的在线业务开发平台，面向全百度的在线业务支撑平台，专注于总结大社区类业务模式，其提供了标准的webserver环境、标准php环境、AP框架、SAF社区业务框架、基础库、RAL资源访问层、KSARCH通用服务等组件，统一业务的逻辑和部署结构，为测试、运维等提供一致的视图。ODP作为百度内部PHP开发的标准框架，覆盖了公司大部分把PHP作为业务开发语言的团队，影响超过千人以上PHP工程师的开发，对RD/QA的学习、开发和测试效率提升100%以上。

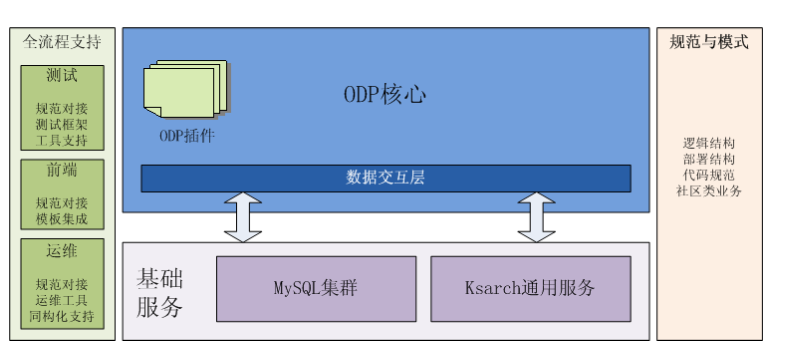


图4-1 ODP缩略图

ODP核心包含了ODP的核心功能组件，包括运行环境、核心基础库、数据交互层、框架等。横向看，ODP核心通过库、框架、工具等集成支持了各类规范和模式，也为全流程支持提供接口。向上看，ODP核心直接为产品线业务提供运行环境和研发支持。向下看，ODP核心通过数据交互层将底层的通用服务提供给业务。ODP核心支持插件机制(参见《ODP组件规范》)，各技术组和产品线可通过插件向ODP增加新功能，比如基础库、测试框架、工具等。

ODP核心通过数据交互层将底层的通用服务提供给业务，目前，ODP为大社区类业务集成了较为完备的通用服务支持。

1.MySQL 集群为DBA 统一运维和提供的MySQL 存储服务，具有高可用、高性能、自动读写分离等工业级特性。

2.Ksarch通用服务Ksarch为大社区类业务提供了大量成熟稳定的通用服务，如反作弊、检索、提交、高性能、kv存储、memcached等。

4.1.2 RAL

RAL是一个支持多种交互协议和打包格式的php扩展。

RAL规定了一套高度抽象的交互过程规范，将整个后端交互过程分成了交互协议和数据打包/解包两大块，可以支持一些常用的后端交互协议，标准化协议扩充的开发过程，促进代码复用。RAL集成了负载均衡、健康检查等功能，让上游端不需要再关注这些繁琐的通用逻辑，同时实现版本可以在性能方面有更优的表现。

4.1.3 NMQ

长期以来，百度社区几大产品线（贴吧、空间、知道、ks等）都独立维护着自己的提交系统。虽然产品逻辑和规模不同，但从实现上来讲，面对的问题相近，解决的思路相似，维护重复、运维分散、人力浪费。

Nmq 是New Message Queue的简称，是一个通用的消息队列系统，为在线服务设计，用于替换并统一UGC（User Generated Contents）产品线的提交服务，由ksarch 开发，维护.简单地说，nmq接收上游提交过来的消息，将它保存起来，另一方面，又将储存起来的消息发送出去。

proxy是nmq整个集群的入口，无状态、无单点，它的作用是将收到的数据发给topic.Proxy将上游收的消息发送给topic，这里topic是单点；这个接收所有proxy发来的消息，并且保存起来的topic称为主topic，它保存有全量数据。在某些应用场景，我们可能需要多份全量数据，所有数据必须一致。所以我们选择和主topic进行消息同步，从主topic同步数据的称为备topic。

4.2 缓存系统

工作中主要使用缓存系统主要为memcached和redis。Redis相对Memcached功能和特性更多； 而对于性能，Redis作者的说法是平均到单个核上的性能，在单条数据不大的情况下Redis更好。为什么这么说呢，理由就是Redis是单线程运行的。 因为是单线程运行，所以和Memcached的多线程相比，整体性能肯定会偏低。因为是单线程运行，所以IO是串行化的，网络IO和内存IO，因此当单条数据太大时，由于需要等待一个命令的所有IO完成才能进行后续的命令，所以性能会受影响。而就内存使用上来说，目前Redis结合了tcmalloc和jemalloc两个内存分配器，基本上和Memcached不相伯仲。如果是简单且有规律的key value存储，那么用Redis的hash结构来做，内存使用上会惊人的变小，优势是很明显的。

在缓存使用上，当服务器内存有限时，如果大量地使用缓存键，并且过期时间设置得过长，就会导致缓存占满内存； 反之，如果为了防止内存过大，而把所有的key的过期时间设置过短，可能会导致缓存命中率太低，降低效率；同时，可能会存在大量空闲内存； 采取中庸之道，在时间和空间上找到平衡点，给每个key都选择合适的过期时间，这样做难度较大。所以我学习到的结论是： 限制缓存能够使用的最大内存，防止因为缓存导致服务器崩溃； 添加淘汰策略，当缓存达到内存使用上限时，通过淘汰策略舍弃/持久化被淘汰的元素。

使用缓存对提高系统性能有很多好处，但是不合理使用缓存非但不能提高系统的性能，反而成为系统的累赘，甚至影响系统运作，产生很大的风险。对于频繁修改的数据、没有热点的访问数据、数据一致性要求非常高的数据，不建议使用缓存。

4.3 MySQL

作为后端开发最重要的技术之一，在学习接触新知识时，进一步加深对mysql的理解应用是非常重要的。如何能发挥mysql的高效，建立最优化的索引执行高效的查询都是非常重要的。MySQL查询性能的优化涉及多个方面，其中包括库表结构、建立合理的索引、设计合理的查询。库表结构包括如何设计表之间的关联、表字段的数据类型等。我们通过响应时间来对查询进行分析，找出消耗时间最长的查询或者给服务器带来压力最大的查询，然后检查查询的schema、SQL和索引结构，判断是否有查询扫描了太多的行，是否做了很多额外的排序或者使用了临时表，是否使用了随机I/O访问数据，或者太多回表查询哪些不在索引中的列的操作。在发现查询效率不高时，首先就需要考虑查询语句的设计是否合理。

第五章 工程计划管控与执行情况

5.1 ××××××

5.1.1 ××××××

……

5.1.2 ××××××

……

5.1.3 ××××××

……

5.2 ××××××

……

5.3 ××××××

……

1. 阐述针对复杂工程问题实施方案的执行计划安排，说明对工程管理原理和经济决策方法的学习与掌握，以及在工程计划管控执行过程中的综合应用。
2. 分析、总结和归纳实习执行过程中存在的主要问题，基于具体案例说明应对项目风险与挑战的能力培养情况。

第六章 职业素养与工程伦理的学习与培养

6.1 程序员的职业道德

6.1.1 为人正直，忠于职守

人们常说：先做人再做事，它强调的是人的品质问题，一个人做事再优秀，但为人方面却有不良问题，相信他也得不到众人的敬佩与赞赏；企业招聘人才，通常很难找到技术能力及各方面素质都完全符合要求的人选，因此很多企业会将很多的考核点放在人员的道德品质及忠诚度上，一个人的技术水平暂时落后并不可怕，只要他具有良好的品质、并且积极进取忠心耿耿，那么他也是一个不可多得的人才，他会通过自已的努力在较短的时间内达到企业的要求。

在辛苦的开发过程中，一点特别铭记在心：尊重他人的智慧。其他如：树立正确的技能观。决不能利用自己的技能去从事危害公众利益的活动，包括构造虚假信息和不良内容、制造电脑病毒、参与盗版活动、黑客活动等。具有良好的工作责任性，不能以追求个人利益为目的，不随意向他人泄露工作和客户机密。做到这些之后，想必内心将是有一番更为深刻的体验，找准自己的定位，然后付诸行动，将知识化为实践。

我所认识的程序员中大多数人都为人正直，忠厚老实，而且通常羞涩内向，无论是做事还是说话，我们都喜欢直来直去，没有城府，不懂拐弯抹角，也不会油腔滑调，可以说我们是一群可爱的好人，我并不是在自吹自擂，很多时候我都庆幸自已在多年之后还能拥有那份纯朴与善良。总而言之, 既然选择了这一行，就好好去爱，忠于软件事业，无论是做一名普通的程序员还是xx经理，都要始终坚持严格要求自己，对同事以诚相待,对工作忠于职守,在一家公司做多久就要认真负责多久。

6.1.2 严守商业秘密

目前在国内的IT行业，我认为最可怕的并不是盗版问题，而是IT人才流失时技术（比如源码、文档等等）也跟着一起流失，这是防不胜防的，有些大公司比如一些外企，做得相对较好，计算机上所有外部设备的接口都被封掉，不得上INETERNET，不得往外发EMAIL或被监控，但不用多说，这种管理方式有利有弊，通常国内的中小IT企业很难做到这一点，因此基本上靠员工的自觉与人品，这就正如心理医生必须为病人保守秘密一样，作为一个程序员或IT人才，当你从事某个产品的研发或接手一个项目时，你得为与此产品或项目有关的东西比如源码、文档等资料保密，这些东西无疑是公司的商业秘密。

6.1.3 尊重别人的劳动

这一点不仅仅是我们这一行的，它应是各行各业一个最基本的职业道德，我们从小就被灌输“五讲四美三热爱”的思想，尽管如此，你在大街小巷还是经常看到乱吐口水乱丢垃圾乱撞红灯的现象，这不得不让我们怀疑中国的教育模式与质量，回到IT行业，泛滥成灾就是盗版问题，没有使用过盗版软件的IT人员真的是极少极少，可能没有，呵呵，在一定的程度上这应不是道德问题，我是一个好人，但有时我不得不用盗版，因为我穷啊，一个软件就花我两个月的薪水，我还用生活吗？有关软件的盗版问题，太多的原因，太多的问题，并非三言两语可以说清道明，我能说的就是：劳动是伟大的，是光荣的，劳动是辛苦的，劳动是要付出心血和汗水的，因此任何人的劳动成果都应受到称赞，任何人的劳动都应受到尊重。

6.2 重视能力的提升

如何做到简单可依赖，我认为是工程师最重要的一环。

web开发技术一更新命的特别快，新技术、新技巧在不断地被发明出来。我们讲作为web工程师的成长，我认为，主要在两个方面，一部分是“能力”，一部分是“知识”。 我个人的观点，能力占百分之八十，知识占百分之二十。

我认为占重点是能力，是非常稳定的，我认为能力是三大块：编程能力、架构能力、工程能力、编程能力，就是用代码解决问题的能力，你编程能力越强，就能解决越复杂的问题，细分又有调试、OS 原理、算法、数据结构、这些的支撑，你才能解决各种麻烦的问题、架构能力，则是解决代码规模的问题，当一个系统足够复杂，你会写每一块，能解决每一个问题，不、接口隔离，也包含认识业务建立抽象模型，也有一些常见的模式，比如经典的 MVC，还有设计层面，面向对象、设计模式、于你能搞定整个系统，这就需要架构能力，架构能力包含了一些意识，比如解耦。最后工程能力，则是解决协作的问题，当系统规模更大，光靠一个人，是没办法完成的，如何保证几个高手互相能够配合好？如何保证项目里面水平最差的人不拖后腿？这个工程化建设，往往会跨越多个业务，以汇报关系上的团队为单位来做。 包括前后端解耦，模块化，质量保证，代码风格。

其实不难看出来，这三项，其实是有顺序的，低、级、小团队，编程能力一项就能应付，越资深的前端，越大的公司和团队，越是需要后面的技能，但是这里其实资深前端，大团队，对能力的需求，是非常看重的，不是说资深的前端，编程能力就可以变差。社区总会拥有一些声音，对工程能力，对架构能力持有一种抵触的态度，觉得比较虚，觉得不需要。实际上以某些人所在的岗位来说，也没错，毕竟公司、团队的状态确实可能用不到，但是以个人成长的角度来看，就是大错特错。

6.3 建立自己的知识体系

作为一个程序员、一名工程师，“建立自己的知识体系”是非常重要的。

第一步，寻找线索。你要了解一个知识，可以先找一本书，看看别人都写了什么，了解这个知识学习路径、拓展性。找一些比较准确的，你可以确定它真的足够全面的资料当作线索。如用wsc的标准、作为线索，我就很有信心同样可能比较适合做的资料，还有一些标准文档的附录，和源代码里的结构定义。

第二步，是建立联系。操作同一组数据，正是面向对象的核心概念，建立对应关系，依据高效原则来面对新知识

第三步，是分类。建立联系以后，我们依据知识之间的联系，进行分类，就可以得到一张图谱，在这个图里面，你就可以非常清楚地知道，哪些知识，是非常重要的，哪些，其实是可以互相替代的，而一旦有你之前没见过的东西，你又能通过把它放到图谱里，来快速理解它，或者找出一些很好的替代方案。

第四步，是追本溯源。当我对一个知识体系的全貌有了概念以后，占了全面两个字，接下来需要确认它的准确性。 很多知识，在社区，会拥有很多的争议，该相信谁呢，这是个问题。 而我的答案，就是追本溯源，去找它最初的讨论和定义。

1. 阐述对软件工程系统的质量、环境、职业健康、安全和服务意识的学习和认识，对职业道德和规范的理解和遵守情况。
2. 评价复杂软件工程的解决方案对社会、健康、安全、法律及文化的影响，阐述应承担的责任。

第七章 对软件工程实践以及软件工程领域发展的认识

7.1 ××××××

7.1.1 ××××××

……

7.1.2 ××××××

……

7.1.3 ××××××

……

7.2 ××××××

……

7.3 ××××××

……

1. 基于毕业设计（顶岗实习）经历，评价软件工程实践对环境和社会可持续发展的影响。
2. 阐述软件工程领域国际发展前沿状况，就本专业的热点问题表达自己的看法。

第八章 结束语

8.1 本文内容

……

8.2 顶岗实习项目课题有待进一步解决的问题及方向

……

8.3 对软件工程实践以及软件工程领域发展的认识

……

8.4 本人毕业设计（顶岗实习）收获及体会

……

在工作总结的基础上，经过分析、归纳，明确结论：

1. 功能、指标等是否实现或达到实习任务要求，对应复杂工程问题是否得到有效解决。
2. 顶岗实习项目课题有待进一步解决的问题及方向。
3. 本人毕业设计（顶岗实习）收获及体会。

参考文献

[1] 啜钢，王文博，常永宇等.移动通信原理与应用[M].北京：北京邮电大学出版社，2002.1-13

[2] 佟学俭，罗涛等.OFDM移动通信技术原理与应用[M].北京：人民邮电出版社,2003年6月.1-21

[3] B. Stantchev, G. Fettweis. Time-variant distortions in OFDM[J]. *IEEE Communications Letters*, Vol. 4, No. 10, October 2000, pp. 312-314

[4] P.H. Moose. A technique for orthogonal frequency division multiplexing frequency offset correction[J].*IEEE Trans.Communication*,Vol.42,No.10,October 1994, pp. 2908-2914

[5] J.-J. Van de Beek, M. Sandell, P.O. Brjesson. ML Estimation of Timing and Frequency Offset in OFDM Systems[J]. *IEEE Trans. Signal Processing*, Vol. 45,No.7, July 1997, pp. 1800-1805

[6] R. Abualhiga and H. Haas.Subcarrier-index modulation OFDM[C]. in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun*, Tokyo, Japan, 2009, pp. 13-16.

[7] Mesleh R Y, Haas H, Sinanovic S. Spatial modulation[J]. *IEEE Transactions on Vehicular Technology*, 2008,Vol.57,No.4, pp. 2228-2241.

[8] D. Tsonev, S. Sinanovic, and H. Haas.Enhanced subcarrier index modulation (SIM) OFDM[C].in *Proc. IEEE GLOBECOM Workshops*, Houston,Tx2011, pp. 728-732.

[9] 李晓峰，周宁.周亮.通信原理[M]。北京：清华大学出版社，1973.22-25

[10] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor. Orthogonal frequency division multiplexing with index modulation[C]. in *Proc. IEEE GLOBECOM*, 2012,pp. 4741-4746.

[11] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor. Orthogonal frequency division multiplexing with index modulation[J].*IEEE Trans. Signal Process*., Vol. 61, No. 22, Nov. 2012,pp. 5536–5549.

[12] Youngwook Ko.A Tight Upper Bound on Bit Error Rate of Joint OFDM and Multi-Carrier Index Keying[J]. *IEEE Communications Letters,* Vol. 18, N0.10,Oct. 2014,pp. 1763 - 1766 .

[13] A.Goldsmith.无线通信[M]（杨鸿文，李卫东，郭文彬等译）.北京：人民邮电出版社，2006.143~161

致谢

本论文的工作是在我的企业指导导师XX老师与院内代管教师XX老师的悉心指导下完成的，……

外文资料原文

A Tight Upper Bound on Bit Error Rate of Joint OFDM and Multi-Carrier Index Keying

II.Joint MCIK-OFDM System Model

We consider a peer-to-peer M-QAM OFDM transmission with Nc sub-carriers

that consists of n clusters of N sub—carriers(i.eNc=nN).A stream of M-QAM

symbols is first serial-to-parallel converted,where every n()symbols are grouped into a vector  and  are used to modulate sub-carriers,as in the

classical OFDM,but it differs from that the modulated sub-carriers are only those of n activated indices,similar to [4],[5].

……

外文资料译文

基于多载波索引键控的正交多路复用系统的误码率上界

二．基于多载波索引键控的正交频分多路复用系统模型

我们考虑一个端到端的M-QAM，Nc子载波的基于多载波索引键控的正交频分多路复用系统有n个簇，每个簇有N个子载波（Nc=nN）。M-QAM的符号流经过串并转换之后每n个符号组成一个相量，是和传统正交频分多路复用一样是用来调制子载波的，但是不同的是只有这n个活跃子载波进行了调制。……

……