

一线大厂的架构模板

尼恩 持续 为大家 研究 3高架构，总结和梳理 **3高组件**底层原理、底层源码，3高架构 **行业案例**。

在社群交流的过程中，发现很多小伙伴，不太会写技术方案、架构方案。

所以，尼恩后续会持续 收集、分享、介绍行业内优秀的 架构方案，

下面是一个来自于 AllenWu 的架构方案，是一线的实操案例，对大家很有参考价值

以下文章来源于 后端系统和架构，作者是架构师 AllenWu

这里，尼恩 作简单的梳理，分享给大家。

同时，对于其中不合理的地方，尼恩进行了完善和优化。

团队的技术方案设计模板

不管我们是做业务开发，还是做基础建设，

必然需要做好方案设计，还需要进行方案设计评审。

团队里边很多同学，甚至有些 **T9 级别的同学**，竟然都写不好一个技术方案

究其根本，主要还是没有形成自己的方法论，

从我个人工作这么多年的经验来看，技术方案设计是可以总结出一套方法论或者说框架套路来的。

为此，我总结出了一套通用的技术方案设计模板（提纲），然后在我们团队内部进行了统一，后面还推广到了整个中心，

大家按照这个模板来写方案设计，绝对让你的领导满意。

详细版-技术方案设计模板（提纲）

相对详细和复杂的版本如下：



- 一，现状
 - 业务背景
 - 技术背景
- 二，需求
 - 业务需求
 - 业务痛点
 - 性能需求
- 三，方案描述
 - 方案1
 - 概述
 - 详细说明
 - 性能目标
 - 性能评估
 - 方案优缺点
 - 方案2
 - 方案对比
- 四，线上方案
 - 架构图
 - 关键设计点 和 设计折衷
 - 业务流程
 - 模块划分
 - 异常边界【重要】
 - 统计、监控
 - 灰度、回滚策略
 - 容灾方案
- 五，部署拓扑
- 六，风险评估
 - 潜在风险
- 七，阶段规划【架构演进规划】
 - 第一阶段
 - 第二阶段
 - 第三阶段
- 八，工作量评估

下面是技术方案设计模板在每一章节的简单说明，用来帮助你理清每个章节大概要写什么内容

一、现状

现状，主要是用来描述当前这个业务(项目)的一些基本情况介绍和相关的背景。

方案设计出来之后，是需要给你的 leader 或者团队其他成员进行评审或者查看，
一般来说，由**技术委员会**来评审

但是别人不可能都和你一样清楚你的项目，

因此首先，你要把你项目的基本情况和背景都说清楚，让大家达成一个共识，

大家站在同一个起点上，才能进行后面的方案评审和讨论。

业务背景

业务背景就是你这个业务的基本介绍，包括但不限于：

- 项目名称
- 业务描述
- 业务难点

技术背景

技术背景就是项目是基于什么样的技术背景下来构建的，

可以从 0 到 1 来构建，也能是基于现有的方案来优化，

但是不管是什么场景，一定都会存在相关的技术背景，因此包括但不限于：

- 现有技术积淀
- 现有架构描述
- 现有系统的整体容量

二、需求

需求，很重要！

不管你的技术有多牛逼，都一定为需求服务的，

不管这个需求是技术需求，还是业务需求，一定都是要为需求服务。

注意：需求，这个需求可以是当下的需求，尽可能 包含未来潜在的需求。

需求越完善，后面会少走弯路。

业务需求

业务需求就是你这个业务具体要做的事情，包括但不限于：

- 业务的功能点
- 要提升改造的功能

业务痛点

- 涉及到的业务痛点有哪些

性能需求

除了业务需求，还要从这个业务需求里面考虑清楚我们满足这个业务之下的性能需求点，

如果一个系统不考虑性能，可能流量一上来，服务就挂掉了。

性能需求包括但不限于：

- 预估系统平均容量
- 预估系统峰值容量
- 可伸缩性
- 其他的一些性能要求点，比如安全性等

三、方案描述

前面把现状和需求说清楚后，终于到了我们的重头戏，方案描述这里了。

一般，需要会有几个可选的方案，

也就是说，尽可能把相关可能的方案都描述清楚，

然后给出你认为的最合适的方案，然后让大家来评审和决策，看是否同意你的意见或者有其他更好的意见。

除非万不得已，最好，不要一只有个方案。

方案1

概述

说明一下方案的核心亮点、核心特色，核心目标，核心优势

比如说：高性能、可扩展、双写、主从分离、分库分表、扩容等。

详细说明

详细说明这里需要图文结合，

包括但不限于架构图、流程图 等。

把你整个方案的架构和模块、细节流程都描述清楚

性能目标

性能一般来说可能包含以下部分：

- DAU：
日活跃用户数量。一般用于反映网站、互联网应用等运营情况。
- 平均QPS：
可以参考淘宝的平均QPS 估算公式，进行估算。
- 峰值QPS：
一般可以以QPS的2~4倍计算；

资源评估

给出方案的基准数据，并按性能需求评估需要使用的资源数量。

按照预估性能需求，预估资源数量

- 单节点并发量
- 单节点容量
- 应用服务器的单节点资源配置、节点数
- 缓存的单节点资源配置、节点数
- 数据存储的单节点资源配置、节点数
- 消息队列的单节点资源配置、节点数
- 反向代理的单节点资源配置、节点数
- 搜索引擎的单节点资源配置、节点数
- 伸缩方式
- 高可用方式
- 监控预警的方式

方案优缺点

列出方案的优缺点，

这个需要通过量化的指标来支撑

方案2

可选的另外一种方案，模板和上面一样。

方案对比

前面给出了多种可选的方案，那么这里就是进行一个简单的对比，

然后，给出自己的觉得最优的方案，并且给出支撑的数据、理由。

有了你自己的决策（倾向）的方案后，

当然，对于最优的方案，就应该提供更有支撑力的细化的方案和数据

四、线上方案

线上方案是对上面你更倾向的方案的更为细致的描述。

架构图

整体架构是如何，把架构图画上

关键设计点 和 设计折衷

把核心的关键点，用自己的 名词系统表达出来

这里有一个要求：整体是完备的、自治的

用来确保你的方案的大体方向是 OK 的。

因为没有有一个方案设计是最完美，方案设计都是逐步演进和优化的，

但是，一定是完整的、系统的、没有大漏洞的

还有，方案设计是要最符合当前的背景的。

业务流程

对于业务的核心场景，弄一个整体流程图、核心流程图出来，

然后分业务场景把各个业务场景的流程图也画出来，并且做好相关介绍

模块划分

模块的划分需要考虑我们架构设计的一些原则，

比如：架构分层、业务分模块、微服务化、高内聚低耦合 等。

然后把每个模块的功能点都说清楚

异常边界【重要】

异常边界是比较重要的，一般情况下，

大部分人都能考虑到正常的处理流程，对于异常的边界考虑的比较少，

但是线上出问题，大部分都是异常情况导致，因此这里非常重要!!!

我们可以通过一个 思维导图 去整理相关的异常边界，

这样有助于自己在实现的时候有足够的把控度，也便于别人去 review 你的方案和具体实现（如 coding）

异常边界需要考虑：

- 涉及到了哪些模块
- 涉及到了哪些流程
- 每个模块、流程出现了各种可能情况的处理是？
- 系统底层原因导致的异常的处理是？

监控、预警、统计

线上运行的项目，一定需要有各种监控，预警、指标统计

可以使用 公司内部的基础的监控外，

还需要从业务内部，实现自定义的一些业务监控和相关技术统计

最终的目标、保证系统的高可用、支撑系统的高可用

灰度、回滚策略

- 如何灰度？
- 如何回滚？

容灾方案

容灾就是当出现 IDC 异常的情况下，怎么容灾，这个可以根据实际情况去考虑。

五、部署拓扑

可以按照下面的方向，去做拓展：

- 线上部署拓扑什么，
- 各层的部署架构是什么
- 多活的部署架构是什么
- 公有云部署架构是什么

六、风险评估

标识所选方案的风险，提出解决此风险发生时候的应对策略，

比如：上线失败时的回滚策略。

潜在风险

- 相关的改动有哪些风险点
- 不兼容点？
- 当前设计方案目前存在哪些问题？
- 潜在有哪些问题

七、阶段规划【架构演进规划】

架构怎么演进

阶段如何规划

每个阶段该达成什么目标

第一阶段

第二阶段

第三阶段

八、投入评估

最后，需要做投入的评估，作为投资回报分析的支撑，包括：

- 物理设备、云设备投入评估
- 工作量评估，

这里需要细化到每个模块、

工作量评估 一般按照时间进行，一定要同时包括开发时间、联调时间、测试时间。

最好比较细化，不要太粗，

比如开发时间可以细化到接口的维度，每个接口的设计分别需要多长时间，

硬核面试题推荐

- [阿里面试题: volatile 夺命 连环炮！答垮了就挂，非常重要！](#)
- [惊呆了: Mysql 普通索引、唯一索引，底层区别竟然 这么大！！（讲大厂必备，赶紧私藏）](#)
- [2023高薪必备: 刨根问底 100 个 Nginx 连环炮，让 大厂 面试官 震惊到 怀疑人生！！](#)
- [架构必备: 10WQPS 超高并发架构的10大思想](#)
- [如何写简历: 2023春招，让简历 人见人爱 的8大绝招 | 附100模板](#)
- [聊聊: 什么是脑裂？照此文作答，面试官献上膝盖，秒发offer](#)
- [场景题: 假设10W人突访，系统能不crash 吗？](#)
- [每天100w次登陆请求, 8G 内存该如何设置JVM参数？来看看年薪100W的架构师，是怎么配置的](#)
- [核心面试题: 为什么新生代要两个Survivor区？一个不行吗？](#)
- [核心面试题目: 什么是索引下推？什么是 MRR 优化？怎么才能更好的为表创建索引？](#)
- [核心面试题目: 什么是 回表查询、索引覆盖、最左匹配原则？说说聚集索引、非聚集索引的区别？](#)
- [面试重点难题: Mysql如何实现RR级隔离时，不会幻读？](#)
- [核心面试难题: Java对象为什么 不一定在堆上分配？](#)
- [核心面试题: MVCC、间隙锁、Undo Log链、表级锁、行级锁、页级锁、共享锁、排它锁、记录锁等等](#)

硬核文章推荐

- [100w人在线的 弹幕 系统，是怎么架构的？](#)
- [峰值21WQps、亿级DAU，小游戏《羊了个羊》是怎么架构的？](#)
- [2个大厂 100亿级 超大流量 红包 架构方案](#)
- [一文搞懂: Java高手必备之 Mpsc 无锁队列 \(史上最全\)](#)
- [一文秒懂: 多级时间轮，最顶尖的Java调度算法](#)
- [一文搞懂: 缓存之王 Caffeine 架构、源码、原理 \(5W长文\)](#)
- [一文穿透: 队列之王 Disruptor 原理、架构、源码](#)
- [细思极恐: Java官方JVM 为啥要叫做 HotSpot JVM？背后的水，不知道有多深！！](#)

硬核电子书

《尼恩Java面试宝典》V30版



长按二维码，点击“**识别图中二维码**”即可查看老架构师尼恩个人微信，
发暗号“**领电子书**”给尼恩，获取下面的 价值10W 以上的宝贵资源：

[👉《尼恩Java面试宝典》](#)（极致经典，不断升级）**全网下载超过300万次**

[👉尼恩Java高并发三部曲](#)：**全网下载超过200万次**

- [👉《Java高并发核心编程-卷1（加强版）》](#)，不断升级
- [👉《Java高并发核心编程-卷2（加强版）》](#)，不断升级
- [👉《Java高并发核心编程-卷3（加强版）》](#)，不断升级

[👉《顶级3高架构行业案例 + 尼恩架构笔记》N 篇+](#)，不断添加

[👉100份简历模板](#)



微信搜一搜

🔍 架构师尼恩