第 3 章 Chapter 3

互联网软件可靠性设计与分析

本章介绍互联网平台软件系统可靠性设计的基本思想与实践：首先介绍软件可靠性设计的重要性及工作职责分工；然后从可靠性角度讨论架构设计的原则和 4 种方法；接着讨论可靠架构模型，对可靠性设计的相关因素和故障模式加以分析；再接着讲述可靠性分配方法、预计架构设计完成后的可靠性水平；最后会讲互联网软件系统的可靠性架构实践，包括业务架构、应用架构、系统架构、基础设施部署架构中的可靠性设计实践和方法。

通过本章，读者能宏观地了解互联网软件架构中可靠性设计的基本思路，学习到初步的架构方法和设计过程，在设计过程中有效地分配可靠性任务并进行可靠性评审。第 2 章讲到了可靠性主要关注的系统架构、部署架构、应用架构，本章会把几种可靠性架构设计落地到具体工作中。每一层架构设计都可以深入讨论，限于篇幅，本章只能简要介绍核心点，关于具体每一类架构详细设计的技术和知识需要读者自行学习。

3.1 为什么要进行可靠性设计

要了解为什么进行可靠性设计，需要先了解什么是可靠性设计以及设计工作对可靠性的重要性。

3.1.1 什么是可靠性设计

架构设计是分布式软件设计中非常重要的一环，设计良好的架构能在很大程度上预防和减少故障。软件系统架构设计的目的主要是实现软件系统的高可用、高可靠、健壮性，

预防软件系统出现故障。目前业界形成了一些与可靠性相关的通用架构模式，如微服务架构、同城双活架构、主从架构等，但它们还不能满足可靠性方面的要求。

软件可靠性设计是指在软件开发和软件持续运行的过程中，在遵循工程原理的基础上，采用专门的技术和方法，制定预防措施、改进设计，从而消除隐患和薄弱环节，减少或避免故障的发生。软件可靠性设计也包括针对性地容错、查错和纠错的软件功能或相关工具设计。容错设计使得软件在可控范围内容忍故障，查错设计能够帮助发现、诊断、定位故障，纠错设计帮助高效消除故障。

传统可靠性研究认为可靠性是设计出来的或者通过设计赋予的。一个产品在上线时（我们也叫 $t=0$）一般是能运行并完成预想的基本任务的，是“合格”的，那么为什么在上线后的某个时候（$t>0$）会发生故障，不能满足用户的使用要求呢？这说明 $t=0$ 时产品也是存在缺陷的，如上线前产品的设计和实现不合理，测试工作有遗漏，某些单元没有满足可靠性要求。产生这样的故障的原因可能是架构设计有问题，也可能是上线验收标准有问题，如果经过评审说明这些环节都没有问题，那么可能是系统设计、逻辑架构设计或者运维部署设计有问题。总之，生产故障都可以归为可靠性设计的问题。

3.1.2 可靠性是设计出来的

产品的可靠性是写代码开发、管理、运维出来的，但首先是设计出来的。设计包括产品初次设计和改进设计，对互联网软件而言，从上线开始就在不断地迭代改进版本，一次设计多次改进是常见的架构过程。这里的设计包括组成软件系统的各个子系统 / 模块之间的关系设计，也包括在运维过程中配套系统、能力的设计。

1. 所有的故障都可归结为设计问题

可靠性应该是产品自身的属性。可靠性相关的能力应尽可能内聚于产品自身，形成可靠性自治能力，减少依靠运行阶段的运维能力。那些无法从内部解决的可靠性问题需要从产品外部进行干预，如监控、运维修复等，这些“外部”能力也需要进行相关的设计。软件设计得越好，后续需要从产品外部进行的干预就越少。

产品为什么会发生故障或失效呢？按照传统可靠性的说法，在工作过程（$t>0$）中是否会发生故障，取决于**产品设计过程**中赋予产品的强度（健壮性、高可用、可靠性等）和产品使用过程中导致故障 / 错误的因素（包括所承受的负载、环境变化或随机事件的发生）这一对矛盾体的博弈结果。基于上述的博弈论可把故障分为两种，具体分析如下。

（1）在设计时对工作负载和故障因素估计不足导致的故障

致错因素是指一切能够导致故障、异常或失效的因素组合，如突发流量、网络故障抖动、死机、误操作等灾难事件。工程师的任务是针对这些因素加强可靠性设计，如需要针对机房被雷电轰击、第三方支付出现问题、程序跟 CPU 出现兼容问题等情况加强可靠性设