CAICT 中国信通院

分布式系统稳定性建设指南

（2022 年）

中国信息通信研究院云计算与大数据研究所

2022 年 6 月

版权声明

本报告版权属于中国信息通信研究院，并受法律保护。转载、摘编或利用其它方式使用本报告文字或者观点的，应注明“来源：中国信息通信研究院”。违反上述声明者，本院将追究其相关法律责任。

编制说明

本建设指南自 2022 年 1 月启动编制，在前期研究、框架设计、文稿起草、征求意见和修改完善等五个重要阶段，均面向分布式系统稳定性领域的技术提供方、产品服务方、行业应用方开展了深度访谈、意见征集等工作。参与编制的单位说明如下：

牵头单位：中国信息通信研究院云计算与大数据研究所；

联合单位：阿里云计算有限公司、华为云计算技术有限公司、北京百度网讯科技有限公司、北京银行、杭州笨马网络技术有限公司、思特沃克软件技术（北京）有限公司、中国农业银行、中国科学院计算技术研究所、中信银行、华泰证券股份有限公司、中国工商银行、上海浦东发展银行、蚂蚁科技集团股份有限公司、中移（杭州）信息技术有限公司、深圳市腾讯计算机系统有限公司、建信金融科技有限责任公司、北京火山引擎科技有限公司、浩鲸云计算科技股份有限公司、南京争锋信息科技有限公司、中电金信软件有限公司、四川省农村信用社联合社、北京同创永益科技发展有限公司、中泰证券股份有限公司、中电云数智科技有限公司、安信证券股份有限公司、北京永辉科技有限公司、京东科技信息技术有限公司、南方电网数字电网研究院有限公司、阳光保险集团股份有限公司、上海钧正网络科技有限公司、北京云杉世纪网络科技有限公司、深圳市金证科技股份有限公司、中国银行、中国移动信息技术中心、招商银行、中移（苏州）软件技术有限公司、天翼云科技有限公司。

前　言

随着分布式成为主流的系统架构设计方案，业务系统的迭代速度越来越快，后端系统架构变得越发复杂，单一节点问题可能被无限放大，大规模分布式系统的稳定性保障能力越来越成为业界关注的重点；与此同时，在技术角色分工越来越细，技术专业化程度越来越深的大背景下，分布式系统的架构特性为其稳定性建设中的架构设计、组织设计等也带来了新的挑战。

稳定的系统是产品提供服务的基本前提，但是当前很多企业缺乏解决分布式架构下的系统稳定性、服务高可用建设相关问题的经验。《中国混沌工程调查报告（2021）》调查结果显示，“较多服务的稳定性相对较差，月事故率差强人意”；线下调研结果提示，SRE 团队几乎都是从零开始摸索稳定性建设，在此过程中存在关键技术的建设路径不清晰、建设思路不明确的问题。

针对上述分布式系统稳定性的痛点问题，本文希望形成一份总体性的稳定性建设指南，从全局角度出发对分布式系统稳定性建设工作进行拆解和分析，力求务实、有效地输出有价值的观点。本指南期待能比较全面的帮助中国企业在分布式系统建设、配套组织、运营机制设计层面进行指导落地，实现国内软件发展向更高目标迈进。

目 录

一、系统稳定性建设概述.....1
（一）分布式系统面临稳定性保障新挑战.....1
（二）政策引导 IT 系统稳定性建设平稳推进.....3
二、分布式系统稳定性建设总体视图.....6
三、分布式系统稳定性建设目标.....8
（一）稳定性建设目标.....8
（二）稳定性评价指标.....9
四、分布式系统稳定性建设模式.....11
（一）架构设计.....11
（二）容量设计.....23
（三）运维方案设计.....28
（四）安全设计.....43
五、分布式系统稳定性建设路径.....46
（一）稳定性建设需求分析.....46
（二）稳定性建设实现分析.....47
（三）稳定性建设活动.....48
（四）稳定性建设工具.....54
六、分布式系统稳定性建设行业特点.....71
（一）互联网业.....71
（二）银行业.....73
（三）证券业.....75
（四）通信业.....76
（五）云服务业.....78
（六）零售业.....79
（七）能源业.....81
七、分布式系统稳定性建设展望.....83
（一）人才、生态、标准亟待关注，多重措施提升稳定性发展水平.....83
（二）顺应时代发展需求，推动稳定性建设进入新阶段.....85

附录 1.....88
附录 2.....89

图 目 录

图 1 运维复杂度示意图 2
图 2 分布式系统稳定性建设总体视图 6
图 3 稳定性建设目标视图 8
图 4 中国信通院“稳保计划” 51
图 5 项目开展前稳定性体检视图 52
图 6 项目开展中稳定性测试视图 53
图 7 分布式系统稳定性度量模型 53
图 8 混沌工程成熟度模型 54
图 9 分布式系统稳定性建设工具关系图 55
图 10 稳定性管理建设架构 56
图 11 可观测能力框架图 58
图 12 变更管理能力建设框架图 60
图 13 容量管理能力建设框架图 61
图 14 全链路压测能力框架图 63
图 15 混沌工程平台能力建设框架图 65
图 16 混沌工程与软件完整生命周期对应图 66
图 17 应急平台能力框架图 67
图 18 容灾管理能力建设框架图 69
图 19 应用多活能力框架图 70

表 目 录

表 1 国内推动系统稳定性建设的相关政策 3
表 2 容错等级设计 21
表 3 系统观测覆盖资源 35
表 4 稳定性风险基准表格示例 42
表 5 安全漏洞类型及防范措施 44
表 6 中美稳定性工具开源情况 86
表 7 稳定性守护者列表 88
表 8 混沌工程实验室成员列表 89

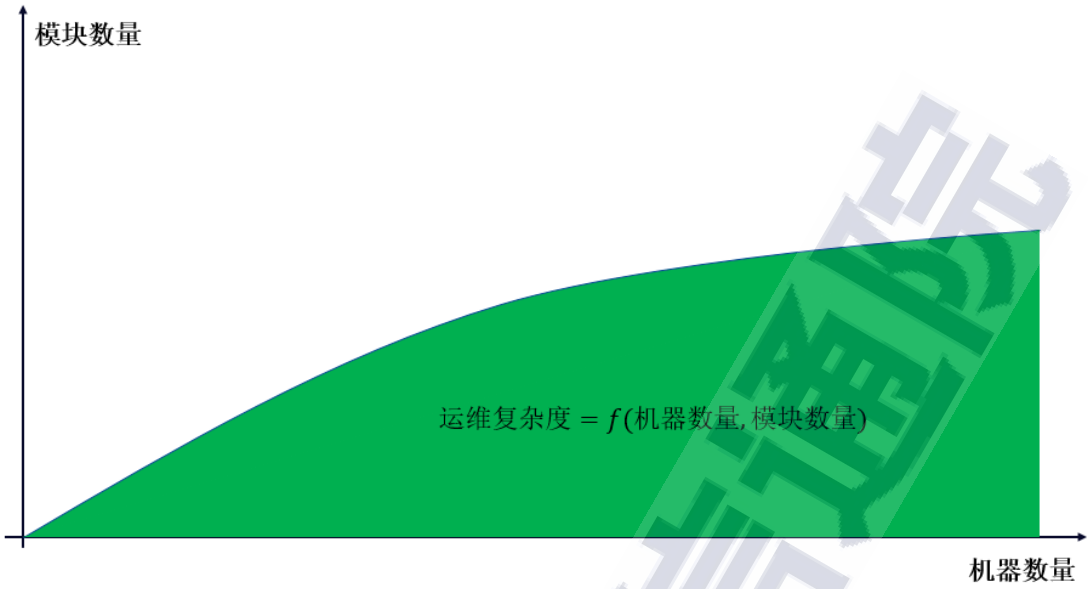
CAICT 中国信通院

一、系统稳定性建设概述

在技术变更、业务挑战加剧以及良好政策引导的背景下，系统稳定性能力建设成为企业等机构组织提升业务连续性能力的核心关注点。系统的稳定性，表示系统在遭受外界扰动偏离原来的平衡状态，而在扰动消失后系统自身仍有能力恢复到原来平衡状态的一种顽性。系统稳定性能力建设是一个系统性工程，需要从企业工程建设的全环节进行设计和实施，充分利用以混沌工程、全链路压测为代表的分布式系统稳定性保障技术，建设保障能力、改造运营流程、推进稳定性文化，保障企业系统稳定性、提升业务连续性、促进行业高质量发展。

（一）分布式系统面临稳定性保障新挑战

在 20 世纪 60 年代，大型主机凭借其超强的计算和 IO 处理能力以及在稳定性和安全性方面的卓越表现，引领了计算机行业的发展，集中式的计算机系统架构也成为了主流。随着计算需求的增长和计算场景的多样化，集中式的处理模式越来越显得捉襟见肘，同时随着 PC 技术的成熟和普及，计算机网络化和微型化的发展趋势在近年不断演进发展，整个分布式计算的理论和实践也走向成熟，计算机系统也开始从集中式向分布式架构的演进。分布式架构在其经济性、自主性、灵活性、扩展性层面较集中式架构有较为突出优势，是近年来各企业进行 IT 系统建设的首选。



来源：公开资料整理

图 1 运维复杂度示意图

分布式系统是由一组通过网络进行通信、为了完成共同的任务而协调工作的计算机节点组成的系统。系统中有大量的服务器及设备，各模块之间存在错综复杂的依赖关系，存在更多的不确定性。整个系统的故障率会随设备的增加而呈指数级增加，单一节点问题可能会被无限放大，日常运行过程中一定会伴随“异常”发生；同时，分布式系统节点分布范围更加广，节点数量更多，物理位置不统一，非常依赖于网络，这对日常运维过程中的日志采集、变更升级等都带来了新的挑战；并且，随着技术角色分工越来越细，技术专业化程度运来越深，分布式系统稳定性落地因其架构特性对架构设计思路、组织设计等带来了新的挑战。

要保障分布式架构下的系统稳定性，需要系统化地探讨稳定性建设新模式。

（二）政策引导 IT 系统稳定性建设平稳推进

为加强企业 IT 系统风险管理，提高业务连续性管理能力，保障国家和人民生命、财产安全，国家对各行业的软件质量及系统稳定性提出了更高的标准、更严的要求，如《中华人民共和国突发事件应对管理法》中要求“完善应急保障制度”；国务院公布的《关键信息基础设施安全保护条例》指出“建立健全监测预警制度、明确网络安全事件应急处置要求”；工业和信息化部发布的《“十四五”软件和信息技术服务业发展规划》中强调“提升软件质量管理能力和软件价值保障能力”，极大鼓励软件高质量发展；中国人民银行印发《金融科技发展规划（2022-2025 年）》，强调高质量推进金融数字化转型；证监会科技监管局组织编写的《证券期货业科技发展“十四五”规划》则强调遵循四项原则，其中第一项为“稳字当头、稳中求进”，等等。由此观之，政策鼓励各行业的研发运维团队培养良好的稳定系统建设观念，在工程设计与实现上规避风险，持续交付高质量软件。

表 1 国内推动系统稳定性建设的相关政策

时间	机构	政策名称	相关内容
2021 年 12 月 24 日	第十三届全国人大常委会第三十二次会议审议	中华人民共和国突发事件应对法	新增“管理体制”一章，修订内容包括：完善应急保障制度；加强突发事件应对管理能力建设。
2021 年 4 月 27 日	国务院	关键信息基础设施安全保护条例	《条例》对制定行业安全保护规划、建立信息共享机制、建立健全监测预警制度、明确网络安全事件

			应急处置要求、组织安全检查检测、提供技术支持和协助等作了规定。
2021 年 11 月 30 日	工业和信息化部	《“十四五”软件和信息技术服务业发展规划》	提升软件质量管理能力。支持配置管理、代码审查、测试验证、质量分析等工具研发，提升质量监控、预警和评价能力。
2022 年 1 月 4 日	中国人民银行	《金融科技发展规划（2022—2025)》	强调高质量推进金融数字化转型
2021 年 10 月 21 日	中国证监会科技监管局	《证券期货业科技发展“十四五”规划》	强调遵循四项原则，其中第一项为“稳字当头、稳中求进”
2011 年 12 月 28 日	中国银行保险监督管理委员会	《商业银行业务连续性监管指引》	商业银行应当将业务连续性管理纳入全面风险管理体系。
2021 年 11 月 26 日	中国银行保险监督管理委员会	《关于银行业保险业支持高水平科技自立自强的指导意见》	坚持风险可控。统筹发展与安全，完善风险控制机制，提升科技金融风险管理能力。
2018 年 5 月 21 日	中国银行保险监督管理委员会	《银行业金融机构数据治理指引》	银行业金融机构应当建立数据应急预案，根据业务影响分析，组织开展应急演练，完善处置流程，保证在系统服务异常以及危机等情景下数据的完整、准确和连续。

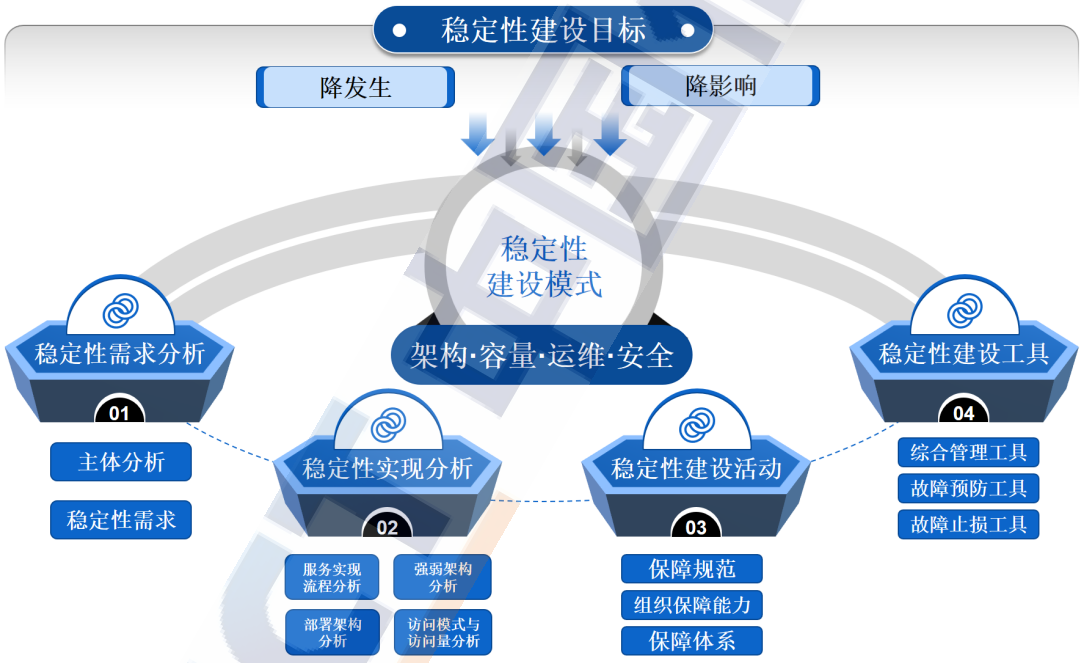
2008 年 4 月 23 日	中国银行保险监督管理委员会	《银行业重要突发事件应急管理规范（试行）》	对商业银行的业务连续性作出了明确要求。

来源：公开资料整理

CAICT 中国信通院

二、分布式系统稳定性建设总体视图

系统稳定性是产品能力的基本要求，保障产品的稳定性，就需要开展稳定性能力建设。稳定性能力建设是一个系统化工程，从硬件到软件，从人员到机制，内容涉及组织内多部门协作、稳定性流程规范制定、体系化技术实现、稳定性文化建设等一系列工作集合。本章将围绕分布式系统稳定性建设模式，结合分布式系统稳定性建设目标，给出分布式系统建设路径，并提出图 2 所示的分布式系统稳定性建设总体视图。



来源：中国信息通信研究院

图 2 分布式系统稳定性建设总体视图

稳定性建设目标：稳定性工作贯穿软件生命周期的全过程，从故障的视角来看稳定性建设的最终目标是“降发生”和“降影响”，稳定性建设目标可以通过评价指标实现量化。

稳定性建设模式：围绕上述稳定性建设目标，我们认为系统稳定性建设思路包括了 4 大建设模式：良好的系统架构和实现、完备的容量规划设计、优秀的运维方案设计，以及规范的安全设计。

稳定性建设路径：分布式系统稳定性建设需要从企业工程建设的全环节进行设计和实施，稳定性需要作为核心考量内建于企业软件工程的全生命周期并佐以建设活动，从流程机制、制度文化层面巩固**稳定性优先**的战略思想，最后通过稳定性建设工具将稳定性保障工作落到实处。

三、分布式系统稳定性建设目标

稳定性建设目标对稳定性建设非常重要，稳定性建设工作的开展都是为了实现最终的稳定性目标，稳定性目标可以通过评价指标实现量化。

来源：中国信息通信研究院

图 3 稳定性建设目标视图

（一）稳定性建设目标

降发生，即降低故障发生的概率。支持应用建设“三高能力”，即高可用、高性能、高质量，从方案设计阶段即采用**面向失败的理念**来设计系统架构，并通过一系列技术手段验证系统“三高能力”是否符合预期。**高可用**，通过冗余设计的思想来实现应用架构的高可用能力保障，同时通过可靠的基础设施组件，来将应用的高可用能力转移到基础设施来提供。**高性能**，通过精简主干业务逻辑，将不相干的业务异步化，实现快速的功能响应；通过缓存技术，加快数据访问的性能。**高质量**，高质量设计更多的是一种软件开发最佳实践经验的沉淀。

通过设立开发、运维的规范可有效减少人为故障的发生；通过合理的拆分理念，如纵向技术分层、横向业务分区的理念，提升系统的可维护性、可演进能力。

降影响，即降低故障发生后的影响范围。早感知，故障感知最基础和重要的原则就是完善监控告警。通过可视化的监控告警能力，感知系统的异常变化，可以尽早发现甚至预测系统故障。**快定位，**系统对故障定位明确，故障管理机制定义完整，职责明确，流程清晰，能有效提升故障发生后的处理效率。**急止损，**“止血”大于“修复”，故障发生后的第一反应永远是“优先止损”，在完成止损工作之后，再开展故障修复。而为了有效止损，需要提前设立各种故障预案。**优改进，**及时复盘，实现故障闭环。故障复盘是故障发生后的改进措施，目的是完成系统韧性提升，实现故障闭环。

（二）稳定性评价指标

稳定性保障是一项非常宽泛且复杂的工作，规划整体稳定性保障体系落地首先需要一组简单清晰易衡量的评价指标来整体牵引稳定性能力的建设。根据企业规模和发展阶段可以酌情从三个维度考虑，评估系统稳定性：业务可用程度、用户影响程度以及资金损失程度。

业务可用程度：业务可用程度是最通常使用的系统稳定性评价指标，即 SLA。通常情况下，SLA 有两种计算方式：一种是通过时间维度计算，一种是通过用户请求状态计算。除 SLA 之外，还可以配合使用 RTO、RPO 等指标，监测数据的完整性。

用户影响程度：稳定性能力建设的目标之一就是降低故障影响，

所以故障发生之后，用户影响程度也是评价系统稳定性的重要指标，这里的影响程度主要是指受影响的用户数量。

资产损失程度：资产损失程度主要从应用方的角度出发，评价故障发生后对组织产生的影响，此处的资产包括有形资产和无形资产。有形资产包括：资金、设备、人力成本等；无形资产包括：社会形象、潜在商机等。

CAICT 中国信通院

四、分布式系统稳定性建设模式

稳定性建设模式是指在开展稳定性建设工作过程中应重点关注的技术方法或方案。在稳定性建设目标的指导下，需要有一系列技术模式来支撑稳定性能力实现，以下就常见的稳定性建设模式进行介绍。

（一）架构设计

根据不同系统业务特点、不同发展阶段（如系统规模、团队规模）、不同系统指标侧重性要求等，有很多不同的架构思路及折衷考量，例如存储选型、服务化治理、中间件选型、中台系统抽象等。本小节会专注于简要介绍会影响稳定性的核心架构设计要点以供读者参考。

1.去除单点

（1）硬件单点

设计确保不存在特定硬件服务器单点。如虚拟机分配上是否存在物理单点、服务是否能够容忍一台或集群中的若干台应用服务器发生故障等。可根据各种崩溃型故障(crash、fail-stop)、阻塞型故障、缓慢型故障、以及任意型故障（逻辑故障等）分别进行分析设计。虚拟机环境下，需要针对物理机与虚拟机分别评估单点问题，原则上所有应用服务器**不前置依赖物理机**。

（2）存储单点

部署架构上确保不存在特定存储主机单点，如服务依赖的数据库主机是否有单点、存储主机部署架构是否存在单点、存储管控节点是否存在单点。

（3） 网络单点

明确网络拓扑中是否存在单点，一般情况下网络基础设施应当是全冗余设计。但如果服务提出了例如专线、防火墙等要求，就必须考虑网络的单点。

（4） 机房单点

对于具备机房级容灾能力要求的服务或业务在部署架构及服务设计上必须考虑不可用 IDC 部署，设计上确认服务对 IDC 部署是否存在约束（如由于专线限制只能存在于一个 IDC 中），提前设计若干 IDC 故障，对服务影响的逃逸或隔离设计。

（5） 基础技术单点

数据服务软件单点：如缓存、文件系统、搜索等数据服务，需要分析服务是否对上述数据服务软件存在依赖。需考虑当上述数据服务软件发生故障时，服务容忍性设计，对于缓存产品防热点设计。

（6） 服务注册中心单点

服务配置中心通常提供的关键能力包括服务发布和订阅推送，消息的发布和订阅关系推送。应用必须明确对配置中心的使用场景，确认依赖关系及注册中心宕机对于自身应用的影响设计。原则上建议应用系统启动和运行时可不强依赖于注册中心，应用必须能够忍受注册中心短暂宕机所照成的影响，亦即注册中心需要具备一定恢复重建或心跳维持能力。

（7） 数据单点/热点

数据库单点规避: 首先需要分析确认应用系统所依赖的数据库故障影响，单个数据库宕机之后是否能够够快速恢复的问题，如果能够故做到快速恢复，恢复的时间大概是多长。重要服务所依赖的数据库，确保数据库主机及存储的单点冗余。应用层面必须做到数据库主机宕机后的快速的 failover 恢复，如行业内通常采用的分库分表设计。

热点数据表单点规避: 对于不同数据库软硬件配置，单数据表的数据量与事务承载能力存在上限，如部分机型单数据表日事务数不应超过 3000 万，否则可能产生表热点。原则上在数据表设计时需确认所有业务数据等级，对于业务流程数据需要强一致性保证的采用强同步机制保证可靠性，若为业务日志数据可考虑采用异步方式缓解对主业务流程数据处理瓶颈压力。

热点数据记录单点规避: 可能形成热点的数据记录有账户、控制数据、锁等。若单数据记录上有事务串行执行要求，则会直接限制系统的最大吞吐能力。另外设计上需保证做到串行操作的锁等待不影响应用及数据库的性能和容量。对于关键服务，在设计之初就必须消除单一数据记录上的并发操作，避免形成热点；对于非关键服务，允许设计之初有数据记录热点，但必须建立处理量跟踪机制，保证在热点实际发生前，提前消除热点；所有服务的数据库锁操作必须明确的定义锁等待的时间。

（8）内部服务单点

内部业务服务是指由自研软件提供的业务服务。原则上不允许有高等级服务依赖低等级服务。对于最高优先级服务提供方，必须尽最

大可能，将内部业务服务单点个数降到最低，包括提供降级服务能力等。对于最高优先级服务依赖的组件，必须保证所依赖的组件发生各种类型故障时（包括崩溃型、缓慢型或任意型）不能整体崩溃。

（9）外部服务访问单点

外部服务是指非自研软件提供的服务。除非与外部服务提供商有严格的 SLA 保障，否则所有的外部服务可视为低保障等级服务。

设计上充分优化将对外部服务的依赖降到最低。一般情况下，应用不应直接使用外部服务，而应通过统一的内部网关服务来使用外部服务。通过前端方式引入的外部服务（如 JS 引入），也必须考虑在其中。对于关键服务，原则上不建议强依赖于外部服务。如果高优先级服务与外部服务具备业务上的直接相关性，比如支付服务依赖于三方支付，尽可能保证单一外部服务的故障不会传播。理想情况下对于一级服务依赖的外部服务应该有备份可供切换，以消除关键外部服务的单点。

（10）前端资源单点

图片、JS、CSS 等静态资源的可用性设计，建议尽量减少重要服务强依赖的静态资源。设计上做到当静态资源无法访问或访问缓慢时，只影响用户体验，但不影响基本功能操作。

2.依赖设计

高等级服务不允许强依赖于低等级的服务或资源（内部服务、外部服务、数据库、基础技术组件等等）。这里的关键是依赖强弱程度的判断:

最强依赖：当所依赖的服务不可用时，服务不可用，且造成系统崩溃。对于所有的依赖，不建议最强依赖。

强依赖：当所依赖的服务不可用时，服务不可用，但系统不会崩溃，且当所依赖的服务恢复后自动恢复。服务只可强依赖于同等级或高等级的服务与资源。

弱依赖：当所依赖的服务不可用时，服务继续可用，但损失一些次级功能。服务允许弱依赖于低等级的服务与资源。

最弱依赖：当所依赖的服务不可用时，服务继续可用，且无任何功能损失。在成本可控情况下，推荐采用最弱依赖的方式。

分布式系统下的各资源依赖，按类型和层次提炼出来会有如下几种分类：**系统启动依赖、基础软件依赖、业务域依赖、数据库依赖、硬件依赖、网络依赖。**

（1） 系统启动依赖

系统启动只允许依赖数据库、应用服务器本地资源(如本地文件)、公共存储，不允许有其它基础技术服务、内部服务或外部服务依赖。消除启动依赖可以支持当发生大规模故障(如机房停电等)后的快速恢复。

（2） 基础软件依赖

基础软件依赖主要包括消息中心以及数据缓存依赖，同时还应考虑系统软件及其第三方包依赖，应用系统若无特殊情况不应依赖底层操作系统或 JVM 特定版本。

（3） 业务域依赖原则

建议上层业务域可以依赖下层业务域，整体的依赖原则受到系统依赖原则的控制，必须首先遵守应用系统之间的依赖原则，而下层业务域不允许依赖上层业务域。

（4） 数据库依赖原则

把数据库按照数据等级进行分级，不同等级的数据库的数据保护和业务连续性保证都不一样。高优先级应用系统不能够强依赖于次优先级的数据库，以此类推各级应用系统不允许强依赖低于自己等级的数据库服务。

（5） 硬件依赖原则

原则上应用服务不应该依赖特定的硬件设施，比如服务器硬件平台的型号（CPU，内存，主板等），网络硬件的型号，防火墙型号等。

（6） 网络依赖原则

应用系统自身的网络的依赖需求：包括跨机房的网络依赖、外网访问、防火墙等。原则上日常态不建议有跨机房的服务调用网络需求(特殊情况如数据复制、容灾等除外)，实现单机房内自闭环。

3.数据保护

数据保护的主要目的是提升数据安全性，业界一般通过 RPO（恢复点目标）与 RTO（恢复时间目标）两个指标进行度量,核心目标是尽可能缩短数据恢复时间（降低 RTO），避免数据丢失（RPO 接近于 0）。

（1） 服务器单点保护

基于本地盘的数据库系统，数据保护采取跨机房异步复制方式，服务器出现不可恢复性故障时存在数据丢失。

（2）存储单点保护

基于单存储(如 EBS)保护的数据库系统，数据保护采取跨机房异步复制方式，存储出现不可恢复性故障时存在数据丢失。

（3）同机房内多点保护

基于同机房多点保护的数据库系统，采取同机房多份 redo 及跨机房异步复制方式，IDC 机房出现故障时存在数据丢失。

（4）同城异机房保护

基于同城异机房保护的数据库系统，采取同城异机房内多份 redo 保护及跨机房 DG，城市出现灾难时存在数据丢失。

（5）异地异机房保护

基于异地多点保护的数据库系统，数据保护采取跨城跨机房数据保护，人类灾难时存在数据丢失。

4.灾备设计

灾备恢复是指在灾难发生后，将系统恢复正常运作的能力，包含部分数据保护技术。当故障或者灾难发生时，可通过灾备技术保证业务不中断、数据不丢失。从灾备技术的发展来看，主要经历了冷备、主备、双活/多活的发展历程。

（1）冷备技术

冷备技术最初是通过将数据放在异地进行备份，解决了应用及数

据单点问题，确保当主环境出现问题时，可利用异地备份还原数据即可，但冷备技术中备机不能提供访问能力，存在资源浪费。

（2）主备技术

主备技术，通过将部分应用部署到备，充分利用备的资源，但受到数据同步限制，无法做到完美一致性，即使启用备的写应用，仍然是访问主的数据，这种架构应对较大范围故障问题存在不足，当发生如机房级故障时无法做到切换；为了解决以上问题，应用多活应运而生。

（3）应用双活/多活

应用双活/多活是以应用为中心的云原生容灾架构，是容灾技术的一种高级形态，可确保当灾难发生时可在较短时间内实现业务流量切换，尽可能减少灾难带来的损失，有效保障业务系统持续稳定运行。应用多活通过将应用系统部署在多个地理节点，综合考虑各地理节点的电力、供水、网络等基础设施的容灾因素，大幅降低区域性网络整体故障和发生不可抗拒的自然灾害时的服务故障以及丢失数据风险；通过支持各部署单元灵活调整业务接入、同时支持处理业务逻辑、同时提供数据存储等方式，确保当某个部署单元发生灾难故障时，只有部分业务受到影响并按需分配到其他部署单元进行处理。

5.弹性设计

（1）故障隔离标准

系统必须具备防止故障从一个系统/组件传播到另一个系统/组件

的能力。故障从一个系统/组件传播到另一个系统/组件通常有以下两种原因。

系统/组件间强依赖：如果系统/组件间存在强依赖，当一个系统/组件发生故障时，强依赖它的组件将无法正常工作。防止强依赖引发的故障传播，通常的手段是将强依赖转化为弱依赖或最弱依赖，比如设置合适的超时、捕获异常、同步依赖转异步依赖、提供备份组件等。

系统/组件间共享资源：如果系统/组件间存在共享的资源（如线程池、数据库连接池、网络连接池、内存区等），当一个系统/组件因为故障耗尽了共享的资源后，所有依赖该资源的系统/组件也都会发生故障。防止共享资源引发的故障传播，通常的手段是对组件的资源使用建立配额体系，或者为重要组件提供专用资源。

（2）访问量控制标准

访问量控制是指服务提供者或者服务使用者对服务资源有效的 SLA 控制，在做访问量控制设计时，需要关注以下几方面:

a. 服务提供者必须给出本服务(包括系统调用服务、页面服务等）的访问策略，包括最大的访问能力、其它访问约束（如参数约束、单账户访问约束等），说明违反服务访问策略的后果。
b. 服务提供者需要对违反服务访问策略的情况，实施管控措施。我们要求所有对外提供服务的系统（如对外服务的网关系统、对外服务的 web 系统等）必须具有防止外部访问过载的能力（即具备限流能力）。
c. 渠道入口系统需要具备能够降级入口服务的能力，确保入口功

能服务在出现异常时，在交易链路的最前段截断异常，防止影响扩大。

d. 服务调用方需要对关键交易场景下的非关键服务访问进行容错设计，常用的手段包括（熔断、降级），确保在非关键服务访问出现异常的情况下，迅速切断该服务访问，保证关键交易成功率。

e. 服务调用方在调用第三方服务时，需要明确外部服务能力，并具备相应手段可以进行访问控制。

f. 原则上所有控制访问量的手段（如限流、熔断、降级）均应具备实时调整的能力，以保证在异常访问下系统的动态性能余量充足。

g. 原则上建议设定统一的 SLA 标准定义，按照标准的 SLA 模型进行访问控制的实现，这样可以确保全站的 SLA 模型和控制模型的一致性。

（3） 服务降级、限流与熔断

服务限流是当负载超出系统/组件的处理能力上限时，可能会造成系统响应时间增加或部分业务失败，需要通过业务限流来防止系统响应进一步严重恶化。比如: 某分布式系统能够处理的最大 tps 为 2000，通过规则限制上游服务每秒调用的 tps，当请求量超过 2000tps 后随机或选择性抛弃一些请求。

服务降级是当出现系统/组件故障后，以牺牲某些业务功能或者牺牲某些客户群体为代价，保障更关键的业务、客户群体服务质量的

应急措施。服务降级可以是人工触发的，也可以是系统自动执行的。所有核心交易场景下的非关键服务访问均应进行服务降级设计，以保证核心交易成功率。如当某非关键的三方支付平台发生故障，可以采取关闭该三方支付通道，确保整体支付成功率。

服务熔断是在分布式系统中避免从系统局部的、小规模的故障，最终导致全局性的后果的手段。它是通过快速失败（Fail Fast）的机制，避免请求大量阻塞，从而保护调用方。比如：一个服务 A 调用当下游 B 超时或失败时，会导致请求超时引起堆积队列，从而导致下游 B 系统压力越来越大而无法恢复。当触发熔断后，到下游 B 服务的压力减小，从而保护了存活的系统。

（4）容错设计

容错设计对系统稳定性至关重要，在容错设计中，首先应确定系统容错等级策略，策略分级可参考表 2。

表 2 容错等级设计

容错设计等级	等级描述
无容错性设计	所依赖的外部资源访问出错，本应用未能检测识别到，导致应用处理数据出错，造成脏数据的
弱容错性设计	所依赖的外部资源访问出错，本应用服务不可用且难以恢复的
基本容错性设计	所依赖的外部资源访问出错，本应用服务不可用，但是由人工操作后可恢复的
较强容错性设计	所依赖的外部资源访问出错，本应用服务不可用，但可自动恢复的
强容错性设计	所依赖的外部资源访问出错，本应用不受影响并正

	常对外提供服务的

来源：中国信息通信研究院

确定好容错策略分级之后，开始系统容错设计。系统需要提供充足的容错机制，以应对所依赖的外部服务或其他依赖资源发生故障情况；系统的设计原则应该本着不信任外部资源（外部服务、DB、网络设备、存储、消息等）100%可用的原则，在关键处理路径上针对上述可能发生故障的点进行容错加固设计，保护系统自身的可用性。

服务不可用容错设计：跨系统服务调用，调用端必须保障请求准确送达、服务端必须保障响应准确返回；基于此原则，某些场景下可能发生请求送达或响应返回丢失的，必须使用重试机制来弥补，如通过异步确保消息通知机制来解决跨系统、一次性调用场景下请求无法确保送达问题。服务提供方系统发布中、或其他不可预知的服务访问超时，都有可能导致客户端请求失败，此时客户端应用若无任何容错机制，则业务处理异常中断。

关键应用的容错设计：某些关键应用服务需要多个系统参与处理，往往需要在多个系统中针对同一类型、性质的风险点进行多重加固，如一次嵌套分布式事务的所有参与者本身都会对主事务号的唯一性进行检查，通过主事务记录、其他参与者自有的事务记录、默认参与者自有的事务记录等多个唯一性约束装置来多重加固。

数据库容错设计：业务在正常执行的时候，如果遇到某个数据库故障，需要具备快速的容错机制，保证后续业务的正常进行。这种容错机制包括数据库的随机拆分方案，数据库的灾难转移或切换方案。

（二）容量设计

容量设计的目的是根据业务优先级、资源消耗情况等合理评估及分配资源，良好的容量设计可以提升核心业务稳定同时带来成本节约。

1.数据增长预测

数据库访问量：计算服务实现中对每一个数据库的访问量，可以表达为每秒/分事务数（TPS/TPM）或每秒/分查询数（QPS/QPM），确保对数据库的访问量在数据库可承受范围内。

数据库数据增长量：计算数据的增长量，增长量可以表达为每日增加的记录条数、增加的数量存储量。确保数据增长量在数据库与存储可承受范围内。

数据库连接数：计算承载服务的应用集群对数据库连接数的需求，确保所依赖的每一个数据库的总连接数不超过数据库的承载能力。

其他数据服务访问量与数据增长量：除数据库之外的其他数据服务访问（如缓存、搜索等），可参考数据库访问量与数据增长量的评估方式进行评估，确保数据服务的访问量与数据增长量在可承受范围内。

2.网络流量

内部网络流量、连接数与请求数：计算服务处理对内部网络流量、连接数与请求数的需求，确保不超过内部网络设备的承载能力。内部网络流量、连接数与请求数需要包含交换机、负载均衡设备、SSL 设备、防火墙、专线等。内部网络流量计算复杂，推荐两种方式：基于

性能测试评测和基于生产服务实际网络流量占比推算。理想情况下可以针对每一个服务的访问量与网络流量之间建立计算公式。

外部网络流量、连接数与请求数：计算服务处理对外部网络流量、连接数的需求，确保不超过外部网络设备（含外部负载均衡设备、SSL 设备、路由器、交换机、防火墙等）的承载能力。外部网络流量的直接计算复杂，建议通过全链路压测评测。

3.消息量

计算服务处理消息量与消息体大小，确保不超过消息中心(或消息队列)的承载能力。若承载能力无法支持，则考虑对消息中心进行扩容支持。如果涉及到跨 IDC 消息投递，需要进行跨机房消息性能和容量的评估，原则上非必要不建议跨机房消息投递。

4.内部资源使用

数据源连接池：在给定的服务访问量下，针对数据源连接数（MAX/MIN）配置预估与设计，合理连接数的粗略估算方法可以通过事务并发数与事务长度来进行。如事务并发数是 100TPS，事务长度是 500ms，则 MAX 连接数至少要大于 50（100TPS*0.5s）。但合理的连接数最好通过性能压测获得，或者根据处理模式与访问量相似的系统的生产数据推测。设置最大连接还必须同时评估对数据库总连接数的影响以及与对 JVM 内存的影响。在无可参照系统的情况下，最稳妥的方式是通过压力测试来验证。同时，严格设定连接池的 BLOCK_TIMEOUT 值，确保系统在故障时期的容量和性能。若服务

瞬时并发量大，建议考虑数据库连接池和预编译SQL语句预热设计，规避应用发布或重启后的瞬间流量冲击。

事务长度：数据库连接在整个事务执行过程被占用，因此长事务对系统容量有严重的负面影响。由于远程调用、底层资源访问、长循环是典型的长时间操作，因此，必须将远程调用、底层资源访问与长循环尽可能从事务中移出，将事务长度降到最低。特别需要注意的是，在事务中进行可能产生阻塞的操作，后果可能非常严重，如无超时的远程服务调用、底层资源访问、可能的无限循环等。对链路比较长的业务，可以通过异步化的方式来减少数据库事务的长度。

线程池配置：在给定的服务访问量下，确定每一个线程池的配置的合理性。线程池过小，将无法足够的并发能力来支撑所需要的服务访问量，线程池过大，则可能超过一个 JVM 可以支持的线程总数，且对性能可能产生负面影响。建议使用有界线程池，避免造成线程数量不可控。建议通过压测对线程池配置进行调优，在满足并发要求下，采用最小的线程池配置。

网络连接池配置：在给定的服务访问量下，需要确定每一个网络连接池配置的合理性。网络连接池过小，将无法提供足够的并发能力来支撑所需要的服务访问量，连接池过大，则对网络设备与目标服务器产生过大的压力，且对性能可能产生负面影响。建议通过压测对网络连接池配置进行调优，在满足并发要求下，采用最小的网络连接池配置。

JVM 配置：评估 JVM 内存配置、GC 算法与参数配置是否合理。通常情况下，JVM 配置可以根据线上参考系统（应用的架构、典型处理模式、访问量与访问模式等）进行配置。但如果应用有特殊需求（比如开设了大的内存缓存、队列，或者对响应时间与吞吐量有特殊要求等），需要进行专门调优，并对调优结果进行性能与稳定性测试。需要注意的是，所有的内存缓存、队列以及内部其它数据结构查询等都必须设置大小上限，并计算或测试当上限达到时不会出现堆内存溢出的情况。一个容易出现的问题是使用内存数据结构作为批量查询或文件处理的缓冲区，当输入数据量过大或者并发度太高时，堆内存就耗尽了。

5.伸缩性

复制性伸缩：应用复制型伸缩是所有应用都必须具备的能力，因为应用必须是无状态的，可以通过在集群中添加应用服务器实例，可以接近线性地扩展集群容量。对访问量大、读写比高、数据一致性要求低的场景，可考虑缓存与读写分离策略，实现数据复制。应用必须具备容忍一定的数据复制延时造成的数据不一致的能力。

垂直性伸缩：应用垂直型伸缩是指按照功能将应用拆分。需要评估应用拆分与业务架构、应用架构、非功能性需求的匹配度以及粒度是否合理。数据的垂直型伸缩也是指按照功能对数据拆分，同样需要评估数据拆分与业务架构、应用架构、非功能性需求以及运维成本的合理性，取得各方面综合最优的方案。

水平性伸缩：水平型拆分主要针对数据，是指按照用户或请求的维度对数据进行水平拆分。若预判未来业务量增长必须通过数据水平拆分才能支撑，需要在早期实现中就实现数据水平拆分，避免未来重构的成本与风险，行业内水平拆分的技术可以参考 TDDL 等技术。

6.IDC 容量

需确保每个 IDC 容量充足，可以应对任何单个 IDC 宕机的容量影响，原则上对于一级服务必须具备多个 IDC 的容量，确保任何一个 IDC 宕机，剩余的 IDC 都能提供 100%的处理能力。

7.链路分析

同步调用链分析：过长的同步调用链对性能、容量、可靠性都是极大的风险，整个处理的响应时间是链条中每一环的处理时间之和，链条中的任意一环出现故障或缓慢，都会造成整个处理缓慢或失败，所有的服务访问量会压到同步处理链条中的每一环，且每一环存在大量的线程等待（阻塞线程资源甚至更昂贵的数据库连接资源）。降低同步处理链路长度的通常做法有：控制系统的拆分粒度，优化系统的职责；对于大访问量的处理（每日千万级或以上），可考虑将远程调用固化成本地处理，牺牲一些灵活性换取稳定性与性能；异步化。

响应时间分析：需要对所有服务的响应时间进行分析。服务处理的响应时间取决于它所直接依赖的内、外部业务或系统服务的处理响应时间，它自身的处理时间、以及请求排队等待的时间。对响应时间的评估不是一个绝对值，而应该是一个响应时间区间。需要找到瓶颈点进行分析与优化，确保响应时间区间满足客户端的需要。

8.吞吐量提升

在实践中，由于资源的有限和调配存在延迟，通过提升系统性能不仅在资源上能降低成本，在稳定性上也有着积极意义，可以降低用户体验延迟，减少机器高负载时不稳定的概率。

基础设施优化：依赖数据库、中间件等通过优化配置、技术创新等针对性能进行优化，例如 gc 调优、慢 sql 治理等。

业务流程优化：降低调用次数、多次网络调用优化批量调用、非核心逻辑异步处理、根据负载自动化削峰填谷设计等。

（三）运维方案设计

系统要考虑持续迭代发布变更以及线上运维的诉求，做到变更可控、系统可观、演练到位。

1.变更管控

变更引发稳定性一般都占据大部分比重，因此需要管理好变更执行过程。

（1）兼容设计

在变更管控各项变更中，如果考虑好兼容设计，其整体的变更就会比较平滑，整个变更的兼容设计会从硬件、软件、数据三个层面展开，其中软件部分还区分基础软件和应用软件，现在从以上部分展开对应的兼容设计需要考虑的原则如下描述。

硬件变更兼容设计。硬件平台变更，原则上不应该影响在其之上运行的应用服务(主机硬件平台升级，网络设备升级，存储设备升级，

防火墙升级），所有硬件升级必须考虑线下兼容性，需要在线下环境进行详细的测试验证，保证生产系统变更稳定性。

基础软件变更兼容设计。任何基础技术和系统软件的升级，原则上不应该影响使用其的应用服务（框架，消息组件，缓存，存储中间件，操作系统，JVM，Apache，JBoss，Tomcat 等），所有基础软件必须考虑线下兼容性，需要在线下进行严格并且详细的测试，保证生产系统变更稳定性。

应用软件变更兼容设计。应用升级方案中应该考虑应用向下兼容能力，无法完全向下兼容的应用升级过程，在联调、预发布及正式上线过程中会引起已有业务服务的不可用，在关键业务路径上的一级服务如果发生不兼容现象后果更加严重，会直接导致变更过程中的大量业务处理中断，引起核心业务下跌。应用可向下兼容的评估点包括但不仅限于：服务接口、方法、入参、返回值及服务方法具体实现的向下兼容性能力；其中服务方法具体实现向下兼容是应用向下兼容能力的最核心表现。对于一、二级关键服务，应用升级过程中必须完全向下兼容，确保发布过程中不产生兼容性问题进而导致业务下跌或其他关键服务不可用。同时服务消费端设计上需要做到客户端可不要求同步升级。

数据变更兼容设计。应用软件系统升级方案往往附有数据存储格式变更，良好的数据兼容性设计对升级后应用平稳上线起到重要的保障作用。数据兼容性设计要求设计方案遵循安全的增量变更原则，即在保障已有的数据存储结构不发生语义变化的前提下，合理增加升级

应用所必须的数据列；并且所增加数据列不对已有业务服务造成影响，如外部系统所调用的查询服务不会中断、业务返回结果不变。原则上，当已有数据存储结构语义发生变化，原存储列所存储值业务含义发生变化时，应该通过新增存储列来完成，避免直接复用已有存储列或修改已有存储列名的做法。

对于重要性高的服务，数据升级后必须完全向下兼容；确实无法做到数据向下兼容的，如原有存储列完全废弃的，应该首先确保外围使用系统业务改造完成后方可上线。

（2） 新版本发布设计

停机性发布。原则上建议非高优先级系统不进行停机发布。对于高优先级系统，应在系统设计阶段尽量避免停机发布，如因系统拆分，数据库拆分，整体架构升级等原因一定需要停机，需严格限定停机范围、停机的时间点与停机时长。如需停机的系统及业务可以独立发布，则除这些系统外，其他系统尽量保障采取非停机平滑发布方式。如因系统耦合度或者业务耦合度复杂无法独立发布，则进行整体停机发布；

发布顺序是否合理。根据系统间依赖指定合适的发布先后顺序。系统发布顺序遵照以下原则：禁止系统启动依赖，无因系统启动依赖导致的发布顺序依赖；对于业务依赖，需保证无相互依赖。高优先级系统原则上不应该依赖于低优先级系统；其他系统默认无发布顺序，可以根据发布进度进行无序发布。

发布时间点。发布时间点需尽量避开业务高峰，尤其是发布过程会对业务产生影响的核心系统。系统发布因尽量避免影响业务，如确

实对业务影响较大又无法在系统设计上避免，需将发布时间点放在绝对业务低峰点。

涉及新旧功能切换。验证切换方案地合理性，可逆性。发布过程中涉及到的新旧功能切换方案，应确保可逆，即切换失败后能及时切回到旧功能。方案需在研发环境进行详细测试，如无法在研发环境进行测试，需在预发布环境进行模拟测试，确保方案正确有效，可回滚。

（3）灰度变更

变更过程需要针对变更影响业务、用户或流程进行必要的灰度设计，以确保变更一旦出现问题，影响在可控范围内。

平台建设部分，对于变更及发布过程，可以通过建立多级验证环境并约束变更逐级变更来提前或在在有限影响范围内发现问题。常用技术环境包含线上和线上多个验证环境，各业务也可根据自身情况选择性进行建设。

线下环境，包括开发环境、联调环境，用于线下研发开发验证及上下游联调功能运行情况。

线上环境，包括预发布、灰度、仿真、线上等多个环境，预发布环境用于开发人员进行线上新功能验证，灰度环境用于引流内部可控用户流量进行持续验证，仿真环境可针对线上流量复制回放验证，最后再生效线上真实环境。

变更灰度过程，在分布式系统中常见通用的灰度过程有 beta 发布、蓝绿发布，进行流量级别的灰度过程，能够满足绝大部分变更灰度验证需求。如果变更复杂度较高或者业务比较重要，在方案设计中

也需要进行更精细变更影响面控制，例如按照影响用户维度逐步生效的设计，但要注意一次业务完整流程中开关一致性问题。

（4）数据迁移分析

发布过程所需的数据迁移方案，需事先在线下环境进行模拟演练，反复梳理迁移过程执行步骤，将可能发生的迁移风险降到最小。数据迁移方案的可行性包括：

方案的完整性：是否本次升级内容所必须包含的待迁移数据项全部覆盖到位。

方案的安全性：对于敏感信息如用户隐私信息的迁移方案，是否存在由于迁移脚本的不合理导致隐私信息泄露风险。

方案的可实施性：包括数据迁移操作方案的合理度（发布过程中完成或者发布前、发布中、发布后多阶段完成），相关角色配合实施步骤，同时必须考虑本项目的数据迁移方案所占用时间是否对同一发布窗口的其他项目造成影响。

方案的可检测性：迁移过程各个阶段的数据完整性、准确性检查脚本是否准备到位。

方案的可回滚性：迁移过程中各个阶段如果发生了计划外风险，必须要终止迁移操作的，是否具备了已迁移数据回滚能力。

涉及重要性高的服务的数据迁移方案**必须完整、安全、可实施、可检测、可回滚**。

（5）可回滚设计

回滚的必要性：应用新版本计划应该制定详尽的回滚计划，能够

在最短时间内将应用恢复至上一稳定运行版本；一般情况下应用本身可回滚，而数据层面的可回滚性是重要的考量因素之一。遵循安全的增量变更原则所设计的数据变更方案具备可回滚能力，发布过程中所产生的增量数据列存储值要求可废弃。

原则上任何应用服务在发布之前都必须具备可回滚的能力，没有回滚能力的系统不允许发布上线。

回滚的复杂性：除应用本身及数据层面的可回滚性考虑外，若服务使用客户端已完成同步升级，则必须考量客户端的可回滚性；极端情况下，若客户端的本次同步升级也造成了其作为服务提供方的使用客户端同步升级，则存在多个应用系统复杂的连带可回滚需求；相关系统也需要评估其应用本身及其数据层面的可回滚能力，作为本次应用升级回滚方案的一并考虑项。

在升级方案设计中，应该提前预知复杂回滚方案的实施成本，防止发生上述的同步升级的多重强依赖关系。回滚方案包括但不仅限于：应用回滚、数据回滚及清理、运维策略回滚、监控方案回滚等。

回滚操作对业务的影响：由于应用升级的回滚实施，必然会影响本次升级业务所服务的业务需求，同时会直接影响对本次升级有依赖的其他业务系统；回滚方案中必须明确本次发布窗口所有相关性需求项目，明确一旦发生回滚处理受影响范围，提前告知相关项目组及业务方，同时尽可能降低多个业务关联性较强项目同一发布窗口的回滚风险。

涉及重要性较高的服务应用升级方案要求必须提供回滚方案，且此回滚方案事先在线下环境得到完整模拟演练并确认可行；回滚完成后要求不得中断服务，业务运行正常。

（6）配置变更控制

涉及生产配置参数的变更，原则上必须进行严格变更审批流程保障。所有对于生产动态配置变更由专业运维保障团队统一操作。

动态配置能力可以从以下方面进行设计：

动态配置变更的时机：预发布变更、发布后变更等；

动态配置的可验证：变更接收方能够以日志等形式验证推送的内容，否则是否推送，何时推送，推送的内容正确与否无法确证；

动态配置的生效同步性：在某动态配置涉及多个系统都需要同步时，应用需要考虑在多个系统间不同步时会出现的问题；

动态配置的容错性处理：防止进行线上配置填写错误时，系统即按照错误的情况运行，动态配置必须有默认值；

动态配置是否系统启动时加载：需要系统初起时加载的内容，需要防止出现系统启动依赖；

周期性动态配置：对于定时刷新缓存方式实现的动态配置，需要保证刷新成功后才更新或者替换缓存内容；不能在主线程中判断和刷新缓存，而应该另起线程刷新，防止刷新缓存出现抖动或者阻塞而影响主线程的功能。

（7）复核验证

每个变更都需要有复核人，对于标准变更，复核人可只对结果进

行复核。对于普通变更和重大变更，复核人需要对变更流程、变更表单、实际操作进行核对确认，对变更后的结果进行日志、监控等检查。复核人应对变更不当而引发的问题负责。

每个变更后，需要有一系列的基于变更清单管理的效果检查的内容。如：服务是否正常启动，功能是否可用，性能是否正常，以及变更的内容是否符合预期。通过对变更效果进行验证，才能最终确认本次变更是否正确。同时，针对服务相关的全局核心指标的监控，在变更期间既不应该出现异常，也不应该随意屏蔽。

2.可观测设计

分布式系统一般涉及较大规模服务集群，复杂度和链路深度较高，因此处于分布式系统各服务节点需要具备完善的日志、监控指标、链路追踪等可观测手段，以便准确观测业务系统运行情况并及时定位处理问题。

对于应用系统观测需要覆盖的资源类型如表 3 所示。

表 3 系统观测覆盖资源

覆盖类型	指标描述
基础设施	操作系统、中间件等运行监控，包括计算、存储、网络资源，如 CPU、load、线程池等
系统服务	链路系统各节点运行情况，便于定位问题节点
应用依赖	系统组件依赖服务，如存储、中间件、第三方依赖
核心组件	应用核心处理逻辑的关键运行数据及报错监控
业务运行	能够直接体现业务运行情况，包括用户体验监控

来源：公开资料整理

此外，系统可观测性数据采集需要设置合理的资源规划，具备应急降级能力，避免数据采集进程占用业务系统过多资源；控制采集数据大小，可采用滚动切割并定期清理过期观测数据方式控制磁盘占用量；避免数据采集影响业务主线程，可采用异步方式输出可观测性数据。

3.演练设计

（1） 预案演练

预案演练主要解决的问题是：根据单个系统的应急预案，模拟应用系统的一种或多种故障场景，验证系统的可靠性。

- 预案演练形式

预案演练根据应急预案组织相关的应急组织机构和人员，利用本地的应急资源和系统环境，针对事先假设的异常应急场景，通过模拟实际决策、指挥和技术操作，完成应急响应及处置的过程，从而检验和提高相关人员的决策指挥、组织协调和应急处置能力。

- 预案演练原则

预案演练要遵循两个主要原则，确保业务能提供连续性服务；案演练范围和风险影响可控

- 预案演练目的

检验预案。通过演练进一步理顺应急处置流程，同时检验应急处置方案的完整性、有效性。

锻炼队伍。通过演练增强演练组织部门、参与人员等对预案的熟悉程度，提高应急处置人员的应急响应效率和应急处置能力。

磨合机制。通过演练进一步检验部门间的应急联动效率，完善相关部门间的工作联动机制。

- 预案演练实践

明确演练场景。明确要演练的故障场景及影响范围。

明确风险和应对措施。提前评估预判各场景演练过程中可能存在的风险，并针对各种风险给出应对措施。将风险和措施告知所有干系人。

明确演练人员。演练人员包括组织人员和参演人员，组织人员负责演练前的策划、文档准备、演练人员与演练环境的落实、演练实施过程中的综合协调及演练结束后的评估总结等工作，以保障应急演练能够顺利实施。 参演人员负责具体演练操作实施。

明确演练技术方案和业务验证方案。演练前检查与业务验证：包含系统检查：检查数据库、负载均衡、应用集群等状态是否正常；应用检查：检查服务是否可用、交易量、交易成功率等指标是否正常；网络检查：检查负载均衡、集群、数据库间网络环境是否正常；业务验证：根据案例进行演练前的业务验证。

切换阶段。明确演练切换的各操作步骤，建议通过工具实现作业编排，自动化执行切换操作。

切换后检查与业务验证。切换后进行技术和业务验证，检查数据库集群、负载均衡、应用集群、网络环境等状态是否正常，并根据案例进行业务验证。

回切前检查。同演练前检查操作，检查系统、应用、网络等状态

是否正常。

回切阶段。通过工具编排操作指令，进行自动化切换。

回切后检查与验证。回切后进行技术和业务验证，检查数据库集群、负载均衡、应用集群、网络环境等状态是否正常，并根据案例进行业务验证。

- 演练实施流程

演练实施流程即演练切换前后每一步操作指令，一般建议三要素形式明确，主要包含：时间，操作，内容。如演练前的操作 0：00 关闭负载均衡，阻止交易进入。

（2） 灾难演练

灾难演练与预案演练的区别首先体现在参与演练的应用范围上，灾难演练是针对整个地区的整个机房发生故障，该机房所有部署的系统全部切换到异地机房的演练，预案演练是针对单个系统的某个或某几个故障场景做的应急预案进行演练。其次是在组织形式上和影响范围上的差别，灾难演练波及的系统范围多，参与人员广，预案演练波及的系统范围少，参与人员少。

灾难演练主要解决的问题是：验证当数据中心整个园区发生灾难，如战争、地震等引起大面积停电，导致整个机房系统不可用的情况下，应用系统如何平稳切换到异地机房启用灾备系统，继续对外提供服务的能力。

- 灾难演练形式

灾难演练根据灾备方案组织相关的组织机构和人员，利用异地资

源和系统环境，针对事先假设的异常灾难场景，通过模拟实际决策、指挥和技术操作，完成灾难切换响应及处置的过程，从而检验和提高相关人员的决策指挥、组织协调和处置能力。

- 灾难演练原则

灾难演练过程中需要遵循以下重要原则：确保业务能提供连续性服务；灾难演练范围和风险影响可控。灾备环境有对等数量的机器部署了同样的应用程序、数据库、中间件等。演练目的就是验证通过切换到灾备环境后应用能正常运行，因此分布式系统的灾备系统建设是灾难演练的前提。建议对分布式系统按照业务重要性建设申请灾备资源，例如重要核心系统申请 1：1 对等资源，非重要核心系统申请 1：0.75 的资源，保证灾备环境的可用性。

- 灾难演练目的

检验灾备方案。通过灾难演练进一步理顺灾备切换流程，同时检验灾备方案的完整性、有效性。

锻炼队伍。通过灾难演练增强演练组织部门、参与人员等对灾备方案的熟悉程度，提高处置人员的响应效率和处置能力。

磨合机制。通过演练进一步检验部门间的联动效率，完善相关部门间的工作联动机制。

- 灾难演练实践

灾难演练方案一般包含如下内容：

明确演练场景。首先主要明确演练场景、演练时间（演练切换时间，异地执行时间，回切时间）；

明确风险和应对措施。要提前判断演练中可能存在的风险和应对措施，将风险和措施告知所有干系人。

明确切换范围及业务影响。其次主要明确参与切换的分布式系统以及其关联系统范围，需要分析切换和回切阶段对业务服务造成的影响，是否需要停机，停机时长，停机影响哪些业务，业务中断时长，是否需要向对监管机构报备等。

明确演练人员。演练人员包括组织人员和参演人员，组织人员负责演练前的策划、文档准备、演练人员与演练环境的落实、演练实施过程中的综合协调及演练结束后的评估总结等工作，以保障灾难演练能够顺利实施。 参演人员负责具体演练操作实施。

明确技术切换方案。明确如何切换技术方案是决定灾难演练成功与否的最关键的一环，在保证灾难演练能够平稳切换，业务影响最小的前提下，明确各分布式系统切换的方式，如使用智能 DNS 切换或通过负载均衡重定向流量等。

关联系统如何配合。在明确分布式系统切换方式后，需要明确不参与切换的关联系统如何接入和启停方案。如配合切换系统关停交易，待切换完成后重启交易。

确保资源充足。切换到异地机房运行期间要根据这期间的业务量预估灾备环境计算资源、存储资源是否够用，否则需要申请资源。

自动化执行。切换尽量采用工具化，对切换作业进行编排流程，实现一键切换。

运行监控方案。分布式应用系统切换后在异地机房的运行需要进

行监控运行情况，需要明确异地机房的监控是否完善，确保切换前后对系统的运行情况始终有监控。

- 确定业务验证方案

组织业务人员在切换前、切换后、回切后等三个阶段，验证分布式系统的业务连通性和业务服务的可用性。

- 演练实施流程

演练实施流程即演练切换前后每一步操作指令，一般建议三要素形式明确，主要包含：时间，操作，内容。如演练前的操作 0：00 关闭负载均衡，阻止交易进入。

（3）混沌实验

混沌实验有相对固定的模式，通常包括实验设计与准备、实施执行和实验结果分析等过程。混沌实验一般通过混沌工程平台实现各类混沌实验的统一管理和执行。

实验设计和准备阶段。主要包括故障场景、稳态指标、靶点管理和实验编排等内容。

实验执行阶段。主要包括故障注入、故障观测、实验防护和故障恢复等步骤。

实验结果分析阶段。主要包括实验报告、问题分析与跟进以及统计度量等。

（4）风险巡检

风险巡检验证方案即可配合上述演练验证方案同步进行，也可独立实施。它是一种白盒化的可扩展风险管理和巡检能力。

一个基础的风险巡检方案包含以下必要的要素：

- 按照分布式系统稳定性各子域进行分类，实现每一个子域下稳定性相关的风险关键数据录入形成一个验证的基准，如表 4 所示。

表 4 稳定性风险基准表格示例

子域	稳定性风险	影响描述	关键指标	修复建议	风险级别	风险评分
数据库	Druid 连接池配置不合理	当连接池配置不合理时会造成数据库操作请求阻塞和延迟。	若 initialSize=0，建议调整； 若 minIdle=0，建议调整； 若 maxActive=8，建议调整	initialSize：初始连接数，连接池启动时创建的初始化连接数量 minIdle：最小空闲连接数，连接池中容许保持空闲状态的最小连接数量，低于这个数量将创建新的连接 maxIdle：最大空闲连接数，接池中容许保持空闲状态的最大连接数量,超过的空闲连接将被释放 maxActive：最大连接数，连接池在同一时间能够分配的最大活动连接的数量	中	5
JVM	线程严重阻塞	严重的锁竞争导致线程阻塞严重，可能对响应时间和 TPS 造成较大影响	等锁线程数（或比例）大于 X	找出等锁线程中不合理的设计进行调整	高	8

来源：公开资料整理

- 实现各子域稳定性指标数据的采集。

不同类型的采集器遵循相同的标准输入及返回值 Schema 定义。可通过 Agent、系统工具、系统命令、虚拟机命令等作为数据采集的媒介。数据采集对象建议以操作系统、虚拟机、应用等为一级域，如：一级域：操作系统，二级域：CPU、内存、网络等等；稳定性风险基准表格中的「子域」推荐与此保持一致。

- 实现采集数据与基准数据的比对，并出具风险报告。

数据比对。将采集到的数据与各子域对应的基准数据进行比对，将命中的数据进行汇总，以报告形式输出。

数据报告。按完成采集验证的子域进行汇总分类，给出整体发现的稳定性风险数量、及风险类型、风险等级分布信息，同时，按子域给出每个子域发现的具体的风险详细信息

- 自动化能力，实现分布式系统稳定性日常巡检。

定时巡检。实现按指定时间周期，指定子域范围的自动进行风险巡检。**触发式巡检。**实现按照特定数据指标阈值自动触发风险巡检。

（四）安全设计

系统安全是系统稳定的基础，没有安全的运行环境，稳定性也无从谈起。系统安全性设计可以划分为如下几个方面。

1.系统设计安全

从系统设计的安全性来说,目前大多系统的分布式结构稍不留神就会产生安全隐患。现在已经有一些代码安全扫描工具（如：Fortify,CxSuite 等）帮助开发者进行一些安全和漏洞识别。常见的由系统设计不当产生的安全漏洞类型及防范措施见表 5。

表 5 安全漏洞类型及防范措施

漏洞类型	漏洞描述	防范措施
输入验证漏洞	嵌入到查询的字符串、表单字段、cookie 、恶意 http 协议头、大文件攻击等。这些攻击包括命令执行、跨站点脚本(XSS)、SQL 注入和缓冲区溢出。	在后台代码中必须验证输入信息后才向服务层提交。
身份验证漏洞	标识欺骗、密码破解、特权提升和未经授权的访问。	程序设计中用户身份信息必须由服务器内部的会话系统提供，避免通过表单提交和页面参数的形式获取用户身份。
授权漏洞	非法用户访问保密数据或受限数据、篡改数据及执行未经授权操作	访问保密数据时一定要根据用户身份和权限来判断操作是否允许。
敏感数据保护漏洞	泄露保密信息以及篡改数据	在储存敏感数据时要采用合适的加密算法来对数据进行加密。
日志记录漏洞	不能发现入侵迹象、不能验证用户操作以及在无法帮助诊断问题	程序设计中状态变更的操作，要记录下尽可能详细的操作信息，以便操作记录可溯源。

来源：公开资料整理

2.部署和操作系统安全

从系统部署及操作系统安全性来说，可以参考以下防范措施：部署的操作系统或系统本身所应用到的组件，需要确保安装或升级相关安全补丁，关闭了所有不需要的系统服务，只对外开放必须的端口，且对访问进行鉴权；定期查看 OS 或引用到第三方组件的安全风险，及时安装补丁或替换升级；定期检查系统日志，对可疑操作进行分析

汇报；应用服务器程序在服务器中文件系统中的目录结构位置应该尽量清晰，目录命名需要尽可能的有意义；应用服务需要创建独自用户，不能以具有系统管理员权限的系统用户运行。

3.数据安全

从数据安全性来说，可用以下的防范措施：对数据库监听地址要有限制，只对需要访问的网络地址进行监听；执行数据备份制度，对存储数据和数据进行定期备份；数据操作授权限制，对表一级及其以上级别的数据库或核心数据的操作授权不应对应用服务器开放。

4.网络安全

从网络安全性来说,可用以下的防范措施：选用企业级网络防火墙；根据具体网络环境，制定尽可能周密的防火墙规则；在外网中传输的数据，应选用合适的加密算法(如:使用 https 协议)进行加密。

五、分布式系统稳定性建设路径

“**从业务来，到业务去**”应当是稳定性保障设计的关键原则，否则再先进的技术也可能只是空中楼阁，脱离实际业务需求，往往于业务产生不了最大实用性价值。在服务业务保障业务持续可用过程中沉淀下来的技术才是最有价值的技术。故而本指南将从软件生命周期、运行周期逐步分解稳定性保障的要点及相关建设思路，从业者可根据自身实际情况选择、规划。

（一）稳定性建设需求分析

在开展稳定性建设工作之前，首先需要确认**分析对象主体**，一般情况下可对以下对象进行稳定性分析：**一个应用系统**，通常以独立的应用系统为分析对象，如聊天软件、交易系统等；**一组应用系统**，通常以业务场景为主体关联，如电商订单支付关联系统、微信聊天关联系统；**一个架构域**，通常一个架构域内应用系统都会有一定的内在联系，以架构域为对象能够尽可能避免可能发生的对长尾业务场景的忽视；当然，从业者可根据实际风险形势，根据实际优先级重点确定核心对象。

其次需要确定被分析对象（如一个或一组应用系统）的**稳定性需求**，确定稳定性需求可采用如下方式：确定被分析对象提供的**所有服务**，包括系统服务、页面表现层服务、restful 服务或者终端设备服务；确定服务的**使用场景**用于哪些业务与系统流程，存在于这些业务或系统流程对应的上游服务，上游服务可以通过服务依赖链路追溯；确定每一个服务的**重要性等级**（一级、二级、三级），一个服务的重要性

等级由强依赖它的最高等级的业务服务决定，根据各服务的重要性等级，确定对象稳定性需求。

（二）稳定性建设实现分析

确定稳定性保障需求之后，进一步采集分析与服务相关的业务、架构、设计、实现、配置、部署、硬件与软件使用信息，只有拥有准确、全面的信息，才能保证稳定性分析结果及稳定性保障设计的可靠性。通常从以下几个方面进行分析（包括但不限于以下各节）：

1.服务实现流程分析

分析明确服务的实现流程，如服务实现的 UML 活动图、UML 序列图或者业务流程图等。

2.强弱依赖分析

分析服务实现流程中所依赖的所有应用系统（以及这些系统提供的服务）。对一个应用系统而言，将它提供的每一个服务所依赖的应用系统汇总起来，可以构成应用依赖总体结构图。

对每一个依赖，需要识别该依赖的以下属性：

依赖强弱：强依赖是指必须的依赖，弱依赖是指可选的依赖；

同步或异步：同步表示需要等待返回，异步指调用发生后无需等待立即返回；

依赖权重：一次服务过程中依赖的次数，即访问的次数。

针对具体的服务类型，需要针对性地开展依赖分析，如：

数据库依赖：需服务实现流程中所依赖的所有数据库，将它提供的每一个服务所依赖的数据库汇总起来，可以构成该应用对数据库的依赖总体结构图。

硬件服务依赖：服务实现流程中所依赖的所有硬件服务，如外部硬件存储，网络（短信，专线等），负载均衡（接入和内部负载均衡机制），防火墙等。

基础技术服务依赖：服务实现流程中所依赖的所有基础技术服务，如消息、缓存、K-V、分布式资源管理等，将它提供的每一个服务所依赖的基础技术服务汇总起来，可以构成该应用对基础技术服务的依赖总体结构图。

3.部署架构分析

架构指系统的顶层结构，稳定性建设工作开展前需分析各个实现组件的生产部署架构，明确系统有哪些部分组成，以及明确系统间的协作关系，如集群划分、集群的大小、集群 IDC 分布、网络拓扑等。

4.访问模式与访问量分析

若访问量、访问模式与业务量之间存在确定的关系，可以给出该服务访问量与业务量之间的函数关系。这样做的好处是可以方便准确地推算出该服务的访问量与访问模式，简化容量分析与规划;

如果访问量、访问模式与业务量之间没有确定的关系，进一步估算合理的服务访问量与访问模式，根据生产实际运行情况持续更新。

（三）稳定性建设活动

稳定性建设模式需要一系列具体的建设活动推进和落地，这些建

设活动涉及人员、机制和文化，全方位的建设活动才能更好地落实建设模式。

1.建设稳定性保障机制

稳定性涉及团队所有不同水平技术人员、所有系统、研发所有环节、线上时时刻刻，单个技术人员是无法保障好的，必须建立团队流程机制来可持续保障。

规范编制。稳定性工作，规范先行。制定**代码编写规范**，规范覆盖日志、配置、多线程、数据库使用等多层面，提升代码质量；制定**变更规范**，提供变更级别、角色职责、活动阶段以及输入输出的详细规定；制定**运维操作规范**，针对公司日志标准，提供统一的日志排查命令及规范，针对运维相关的监控告警制定告警处理流程、告警升级机制。

方案评审机制。在系统的负责团完成系统的建设或改造方案初稿后，需通过由业务、技术、测试、运维领域专家组成的专家团队方案评审，才能进一步对方案进行实施。

测试准入准出机制。进入稳定性测试阶段，要严格审查系统是否达到测试准入条件，即满足测试实施的所有必要条件，如果未满足，则不开展稳定性测试。在稳定性测试实施结束后，严格检查所有测试准出条件是否满足，如：没有进行中的缺陷等，否则不予测试通过。

值班及责任判定机制。设置值班制度，每天有技术人员负责值班，值班周期内的所有问题由值班人员治理，不能及时完成的，添加到 BUG 定期跟踪并统计。在出现生产事件后，由专家团队对该问题进

行详细分析，确定问题的发生原因、解决办法后，对该问题进行问责，明确责任团队、责任人、责任承担比例等内容。避免在稳定性治理中产生“囚徒困境”。

能力考核机制。通过考核机制，提升技术人员对系统稳定性建设的积极性，如在测试阶段，发现系统存在较为低级的稳定性缺陷，则对响应负责人扣除部分考核绩效等。系统稳定性经过验收上线，很好的满足了业务的需求，则增加相应绩效。

故障管理机制。故障管理机制包括规范管理故障响应流程、故障升级机制、故障复盘机制，规范技术人员在应对突发故障时的操作流程，明确职责边界，提升沟通效率，推动故障闭环，提升故障处理效率。

2.建设组织保障能力

企业在应对重点项目的稳定性建设时，会从各方面给予保障支持，包括但不限于以下三个方面。

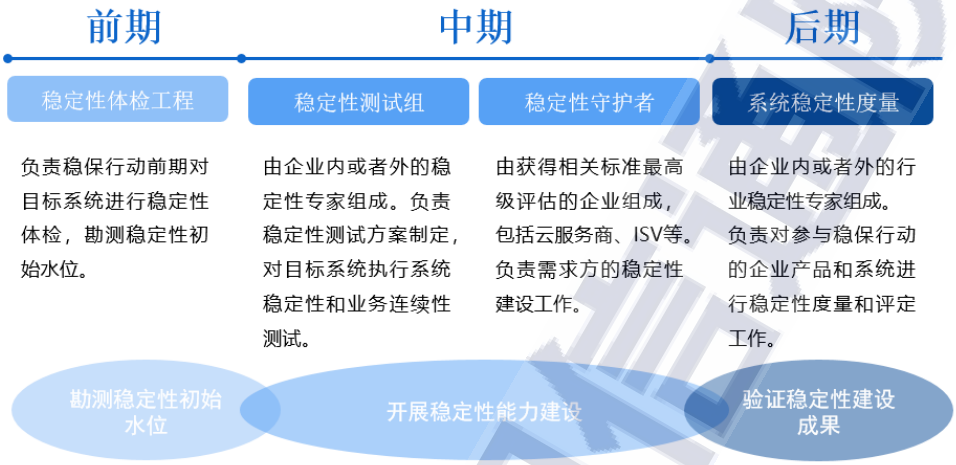
人力支持，从各领域抽调专家进行专项支持，并在人力紧张时，立项招聘相应人员进行支持。

资源支持，提供充足的技术资源支持，如：长期 1 比 1 测试环境部署等。

组织优化，稳定性保障涉及多个团队，容易产生责任划分不清的问题，合理的做法是引入横跨业务线的稳定性团队来干预，调动全公司资源（包括但不限于技术、法务、合规）推动稳定性技术升级。

3.建设稳定性保障体系

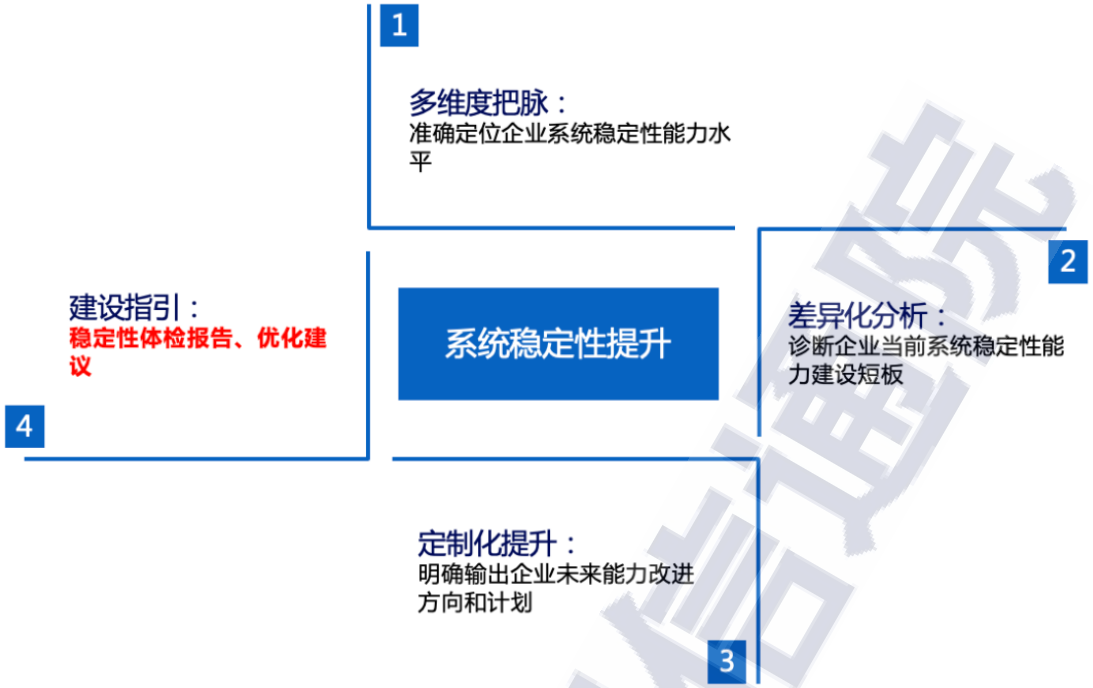
参考中国信通院提出的**“稳保计划”**，在稳定性工程建设前期、中期、后期等不同阶段设置稳定性体检工程、稳定性测试、系统稳定性度量评估环节，全方位推进企业系统稳定性能力建设。



来源：中国信息通信研究院

图 4 中国信通院“稳保计划”

稳定性项目实施前，开展 “稳定性体检工程”，对参与体检的系统进行“望闻问切”，对即将开展稳定性建设的系统进行“望闻问切”，勘测当前系统稳定性水位，结合度量结果给出**稳定性建设路径**，企业可依据体检结果以及需求/预算/时间等制定建设方案，有针对性的提升系统稳定性。



来源：中国信息通信研究院

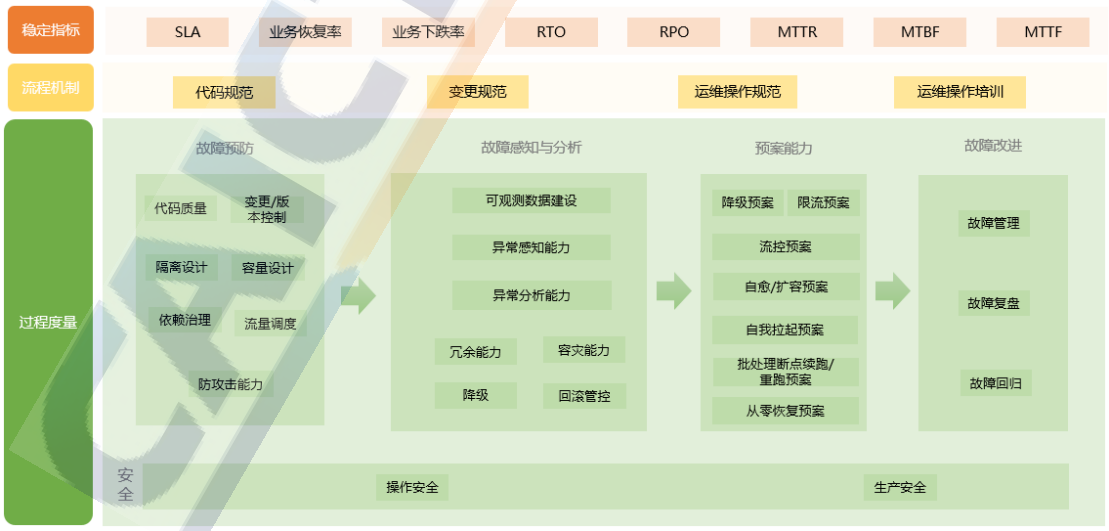
图 5 项目开展前稳定性体检视图

稳定性项目实施中期，依据《混沌工程平台能力分级要求》、《可观测性平台能力分级要求》、《全链路压测平台能力分级要求》、《应用多活平台能力分级要求》等标准开展能力建设与演练测试工作，根据测试情况校准建设路径。并通过参与行业稳定性领域专家组成的“稳定性测试组”讨论制定的云服务与系统稳定性测试方案，对被测系统开展多场景稳定性测试，测试结果校准建设方向、评估行业水平，获取相关证书以自证稳定性能力。稳定性能力建设提供方可参考附录 1 中的“稳定性能力守护者列表”。

来源：中国信息通信研究院

图 6 项目开展中稳定性测试视图

稳定性项目实施后，可以依据“系统稳定性成熟度定位方法”、“系统稳定性度量方法”等开展系统稳定性度量与能力建设的验收工作。通过量化建设成效，验收稳定性建设成果、校验系统稳定性建设成效；同时在成熟度定位方面，为企业进行系统稳定性成熟度定位，如：从工程熟练度、应用成效度和组织建设度三个维度评价混沌工程能力建设后的成熟度水平。



来源：中国信息通信研究院

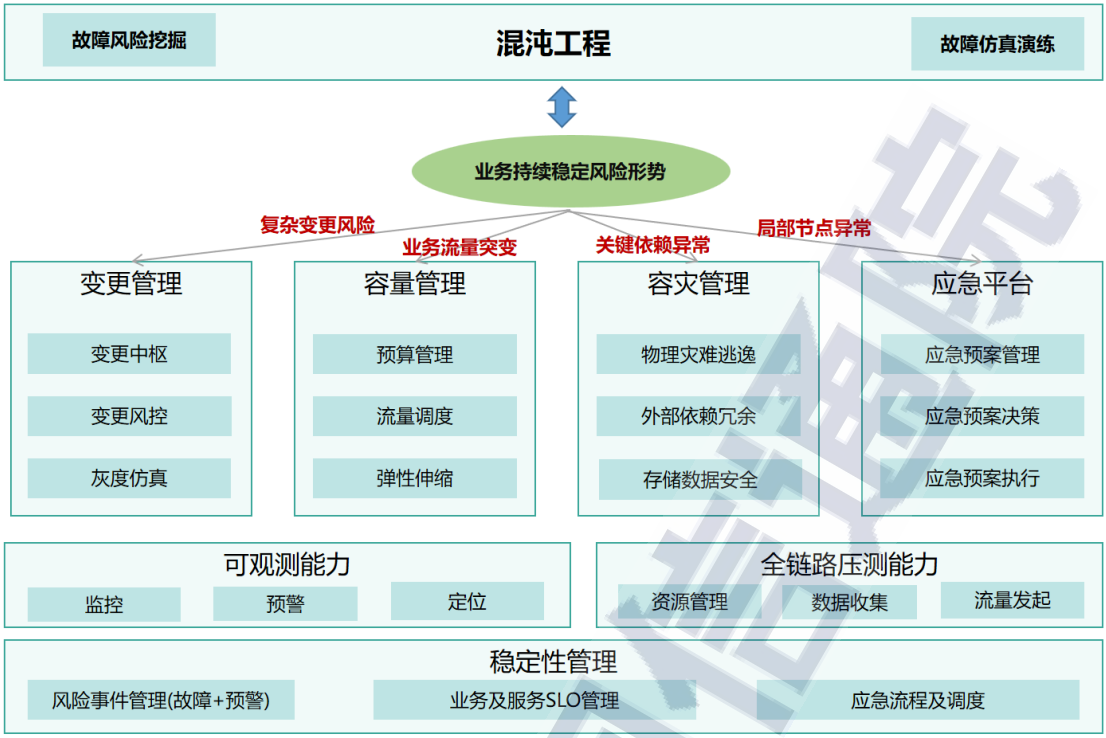
图 7 分布式系统稳定性度量模型

来源：中国信息通信研究院

图 8 混沌工程成熟度模型

（四）稳定性建设工具

稳定性保障能力建设是项体系化工程，本文尝试结合系统稳定性面临的风险形势进行分解，试图从全局视角回答各项关键能力对保证稳定性的技术逻辑。如下图所示：



来源：中国信息通信研究院

图 9 分布式系统稳定性建设工具关系图

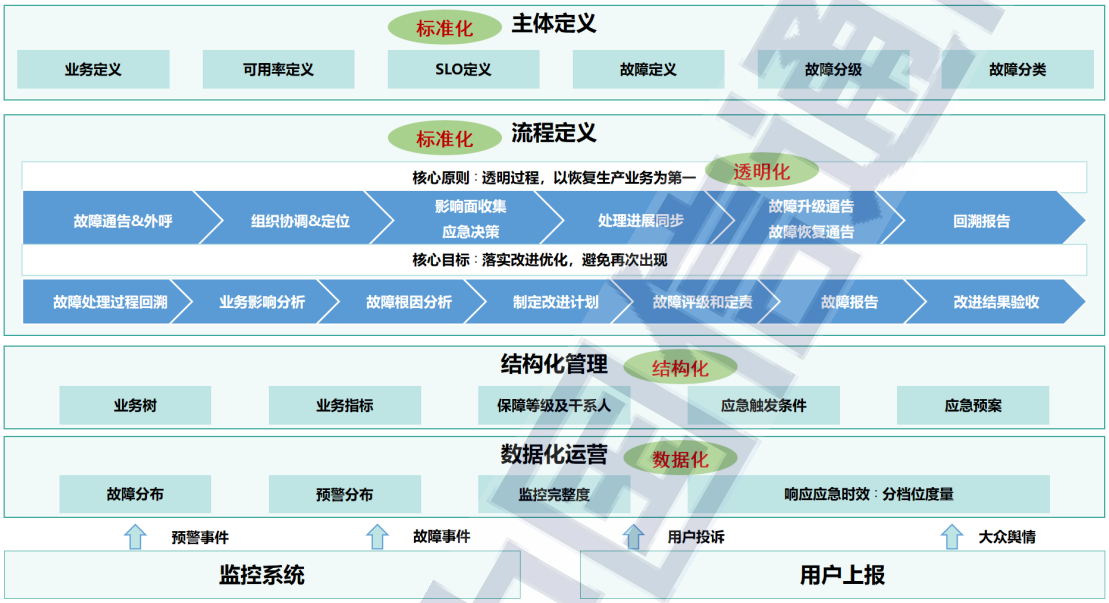
可见这是一项非常庞大而复杂的工程，体系的落地非一朝一夕可完成。故障总会发生，当然也“没有任何一项技术或者平台能够绝对规避风险”，需要通过不断补全完善体系中需要的能力来最大限度降低故障发生概率，或者提升故障应对速度。

对于稳定性保障从业者而言，建议结合业务发展不同阶段所面临的关键风险形势进行规划，拟定合适的建设优先级及实施路径。

1.稳定性综合管理

微服务化日甚的当下，故障影响往往是复杂多样的(单一节点故障可能导致全线业务出错)，往往需要多个技术团队的协同保障系统稳定。需要**统一的系统化稳定性管理能力**作为“连接器”实现多团队协同“透明化”作战，并进一步通过故障应急过程及结果数据复盘，

“数据化”风险趋势以确定建设重点，“标准化”故障管理流程以提升故障管理效率，定义业务或服务的 SLO（Service Level Objective，服务等级目标）以“结构化”组织稳定性保障能力。图 10 示意了稳定性管理能力建设思路。



来源：中国信息通信研究院

图 10 稳定性管理建设架构

a. 标准化：原则、流程及基本定义标准化，形成一致认知的标准，统一管理。

b. 透明化：处理过程信息透明即使传递，避免故障态势由于信息不透明导致错误应急实施反而引发次生故障；并且结合详尽的过程及结果数据，也能够为事后准确复盘、追溯、跟踪等提供更为丰富的信息。

c. 结构化：结构化尤为重要，结构化沉淀的业务树、指标、SLO 等能够通过真实故障不断锤炼优化、并且也能够持续沉淀，会

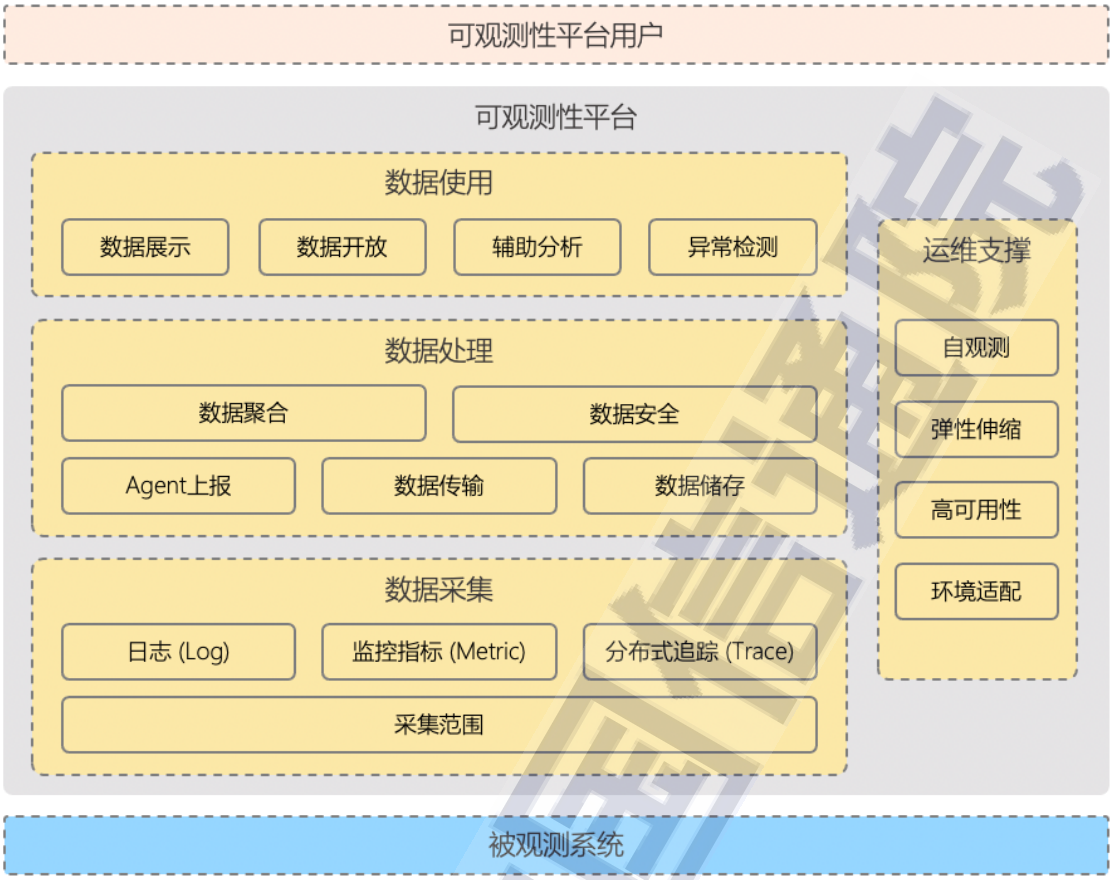
为后续能力演进打好非常好的基础，如结构化 SLO 对 AIOps（Artificial Intelligence for IT Operations，智能运维）演进意义明显。

d. 数据化：可以将“预警响应耗时”、“故障恢复耗时”做为引领性指标，结合真实故障、预警时间建立指标数据化运营能力，驱动各业务技术团队建设其架构自治的故障容忍、应急响应能力。

2.故障预防工具

（1）可观测能力

建设可观测能力的目的是观测业务系统运行情况，并尽早发现故障，定位、分析故障。其中涉及三方面能力：系统运行情况采集、故障发现、以及故障辅助定位分析。



来源：中国信息通信研究院

图 11 可观测能力框架图

系统运行情况采集。主要通过完善监控覆盖率来达成，包含以下几个方面：

a. 数据完备程度

可观测性数据形式上可分为三大类，日志、监控指标、分布式追踪。内容上分为系统数据和业务数据，系统数据包括 CPU/内存负载、磁盘 I/O、网络等，业务指标包括业务成功率、响应时间、吞吐量等。除直接采集之外，也应该允许用户自定义数据采集，或由协议拓展收集数据。

b. 采集范围广泛

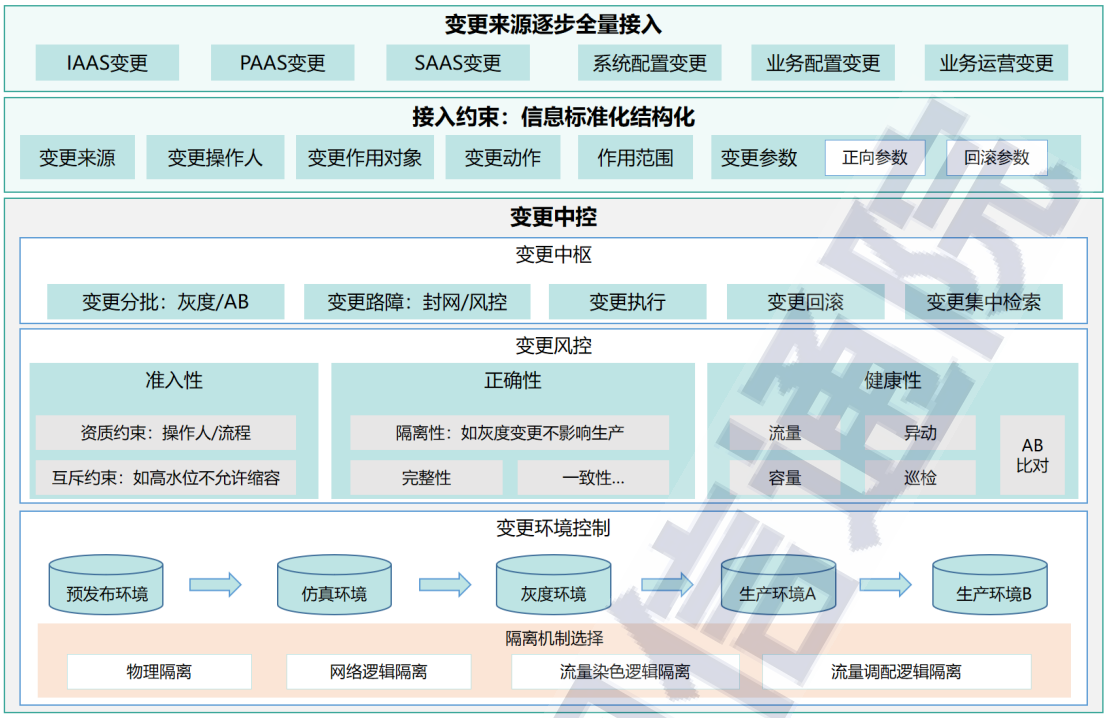
采集范围与具体业务形态息息相关，逻辑上可以从技术层面和业务层面来拆解。技术层面覆盖全局、园区、应用、集群、服务、节点各场景；业务层面覆盖各个产品场景，如网页端、客户端、移动端应用、小程序等。

故障发现主要通过提升监控告警时效来达成。在系统运行情况采集完整的基础上，对可观测性数据进行监控。根据用户输入的规则，及时发现异常数据，并产生告警。同时也可通过对业务重要性分级，提升重要业务告警优先级，如：重要业务 N 分钟告警；非重要业务 M 分钟告警。

故障辅助定位分析。主要通过完善单节点和全链路的故障定位能力来实现。单节点的定位能力，包括 SLO、错误码、热点、内部资源等；全链路的点位能力，包括定位到问题节点、定位到节点根因。

（2）变更管理

对于大多数的变更类故障来说，恢复业务第一要素就是定位到准确的变更并执行回滚。但在分布式架构下，变更源极度分散、变更信息可能也没有形成统一的标准化结构，一方面导致无法以一种架构化平台化能力统一管控全局变更的稳定性能力，另外一方面由于缺少平台化的约束在应急态时变更信息检索效率低下，还有可能定位到风险变更而缺少回滚能力的设置导致故障无法被快速消除。因此，建议可以形成：变更信息标准化、变更中枢统一、变更风控三层能力。图 12 示意了变更管理能力建设思路。



来源：中国信息通信研究院

图 12 变更管理能力建设框架图

（3）容量管理

容量管理的目的是在恰当的时间以一种经济节约的方式为数据处理、存储和计算提供所需的容量。构建基于分布式云原生系统的智能容量平台，主要涉及 3 大类能力：**容量需求管理，容量供给管控，容量调度管理**。

容量需求管控：主要分为资源运营和容量运维能力。资源运营能力包含预算，额度，库存，计划，规划等功能，容量运维能力包含日常运维，临时使用，活动容量，紧急容量等运维能力。

容量调度管理：主要包含容量决策，容量画像，资源核算，性能回归，容量弹性，容量风险管理能力，其中容量决策能力包含对应用容量进行分级（成本、稳定性、标准）和容量运维监控，基于分级进

行的统一决策能力。容量画像包含周期性，流量预测，流量突增，容量多指标弹性和大数据中台和算法实验室。资源核算：包含对应用进行计量和计价账单，驱动应用在稳定性和成本实现最佳平衡。性能回归能力包含制定定期压测和性能环境环境。容量弹性包含水平弹性伸缩，垂直弹性伸缩，对应用涉及到的运维参数进行调节（包含线程池，JVM 参数）。容量风险能力包含限流，自适应限流，流量调度，预案，隔离。

容量供给管控，主要能力是基于资源调度分级，实现在线和混部，对应急资源，云资源进行多重 SLO 和优先级保障。

来源：公开资料整理

图 13 容量管理能力建设框架图

（4）全链路压测

随着压测技术和手段的不断发展，从最初的线下单系统、单模块以及短链路压测，逐步向生产有限条件下压测、生产全链路压测演进。2014 年初，生产全链路压测的方法开始诞生，其目标是希望在大型促销活动来临前，可以在生产环境上模拟路演进行验证整体容量和稳定性。由此，出现了全链路压测方法所涉及的公网多地域流量模拟、全链路流量染色、全链路数据隔离、全链路日志隔离、全链路风险熔断等关键技术。通过全国各地 CDN 节点模拟向生产系统施加压力，通过染色与隔离手段防止压测流量对真实业务产生影响，并在压测过程中对生产系统健康度进行实时监控，快速识别压测对生产业务带来的风险，立即作出流量调节或熔断决策。压测结束后，根据压测报告执行精准的容量规划，保障分布式系统稳定性。

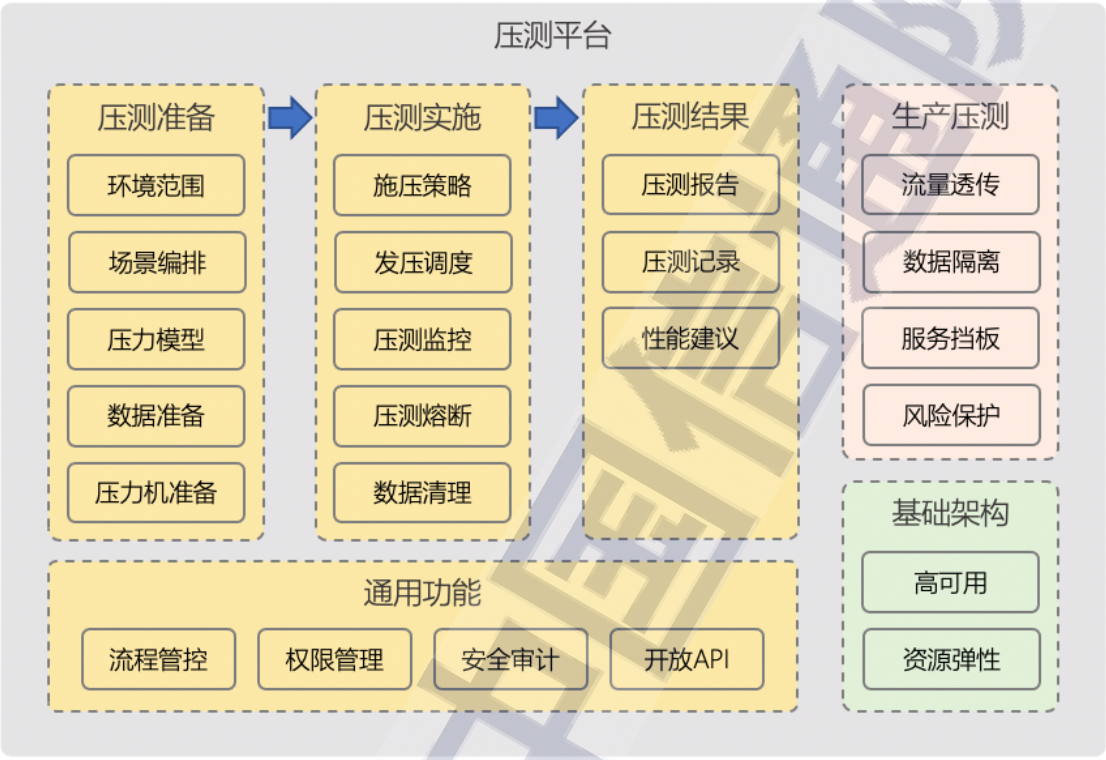
全链路压测常见误区：

a. “有了生产全链路压测就不需要做线下压测。”性能测试的阶段越早，成本与风险越低。线下压测是无风险发现程序级问题，线上压测是低风险实现线下补足和评估生产的容量。

b. “有了生产全链路压测技术，就可以在生产随意实施压测。”生产压测需要充分考虑连带影响，除了防范对业务数据本身的污染风险外，还需考虑对监控、数据分析等下游环节的影响。同时生产压测本身是高风险的行为，所以压测前中后的生产稳定性风险防控能力也至关重要。

全链路性能测试解决了单点性能测试无法从业务的全生命周期

来观测被测系统是否能够满足业务在实际生产中的性能需求的能力。只有被测系统满足了全链路性能测试指标，才能说明系统在线上未发生故障的情况下，有能力进入稳定状态，是系统稳定性的基本要求。



来源：中国信息通信研究院

图 14 全链路压测能力框架图

全链路性能测试能力的构建，主要由以下几部分构成：

a. 资源管理能力，对测试环境资源、测试资源（压力机）等资源进行管理，实现资源的动态管理和充分利用。

b. 数据收集能力，全链路测试过程中，需要对相应的测试数据进行收集，如测试环境数据、业务数据等。

c. 流量发起能力，即从全链路测试环境的起始端，模拟真实用户发起业务流量，常见的流量发起能力如下：

压测工具，通过脚本模拟业务流量，如：JMeter、LoadRunner 等。

流量回放，复制线上的流量数据打入到测试环境中，从而答达到回放效果。

d. 数据分析能力，通过对收集后的数据进行分析，可以得出与测试指标相对应的数据，从而判断测试结果是否满足指标。

e. 结果管理能力，对压测结果覆盖的测试报告、测试方案、测试脚本、测试系统信息等内容进行管理，便测试例复用和测试数据分析等工作。

f. 若需在生产环境中进行压测，则需要被测系统做出相应改造，以规避压测带来的风险。改造内容包含上文提到的流量染色、数据隔离、限流保护等。

（5） 混沌工程

混沌工程概念最早由 Netflix 提出，是一种提高技术架构弹性能力的复杂技术手段，在稳定性验证方面起到至关重要的作用。通过开展混沌工程实验，模拟随机的基础设施层、业务层等各个层面的故障，联合观测系统表现来验证分布式系统的稳定性和可靠性，尽早发现系统潜在的问题，为提高分布式系统稳定性提供参考和建议。

开展混沌工程实验可分为以下步骤：

选定假设设计故障实验：在用户系统中模拟故障效果的发生从而进行稳定性实验的设计过程，对一次完整的实验过程进行的详细流程描述，比如：何时开始实验、发生哪些故障、故障间的先后次序、故

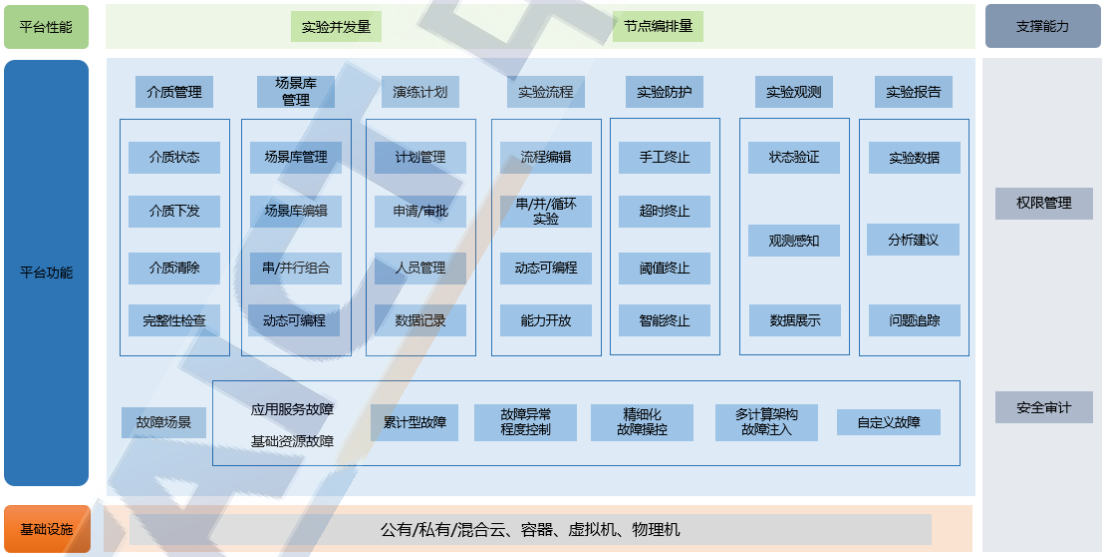
障效果的具体表现、故障持续时长、发生频率等。

自动化编排与执行实验：在系统中构建自动化的编排和分析，包括故障注入与故障释放。故障注入是在用户系统中开始进行故障效果模拟的动作实施。故障释放与“故障注入”相对应，在系统中停止模拟故障发生的动作实施，可根据需要随时终止实验。

故障爆炸半径控制：精准控制故障模拟时的影响范围（故障爆炸半径），以降低混沌实验的风险，避免对生产环境造成较大程度的影响和损害。

实验观测：实验过程中实时观测业务稳态指标表现，包括基础监控指标和业务层特性指标等，据此度量混沌实验的效果。

分析实验结果：对整个混沌工程的全流程进行复盘分析，总结实验结果，进而优化系统稳定性建设。



来源：中国信息通信研究院

图 15 混沌工程平台能力建设框架图

在上述混沌工程基础能力之上，为了让混沌工程在分布式稳定性

保障能力方面充分赋能，还需要在面向软件完整生命周期、面向智能化、面向度量和运营能力体系建设三个方面进一步加强。基于混沌工程为业务基础架构管控、基础设施技术风险和业务技术风险治理提供了强有力技术支撑，促进稳定性保障能力体系建设。

面向软件完整生命周期：混沌工程可面向软件完整生命周期，即开发、测试、发布、运行各阶段，发挥质量守护作用。

来源：公开资料整理

图 16 混沌工程与软件完整生命周期对应图

面向智能化：智能化是混沌工程发展方向和核心主题，通过智能化和自动化技术不断提高混沌工程效率，降低学习成本。例如，通过 AI 技术实现稳态自动化对照分析（系统基础指标、业务指标、监控指标、采样指标等），可以有效提升混沌工程架构感知、智能演练场景推荐、根因分析、安全防护等方面的智能水平。

面向度量与运营能力体系建设：结合 DevOps 体系进行常态化持续集成演练，实现自动风险挖掘和度量，揭示研发质量、资金安全风险、高可用能力、监控预警、应急预案等风险水位，有效促进系统稳定性保障能力系统建设。

3.故障止损工具

（1）应急平台

应急平台的建设主要考虑以下方面：

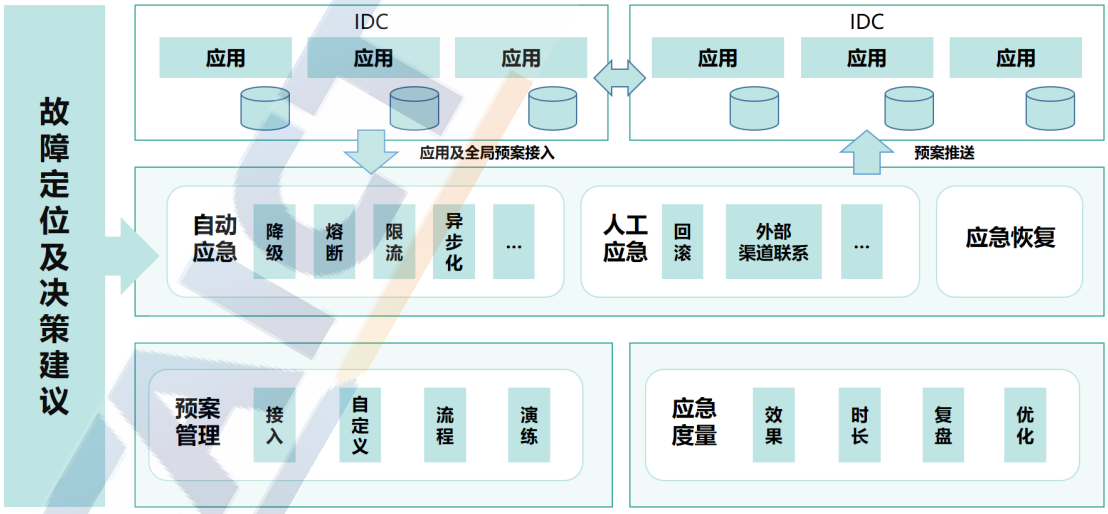
应用设计：应用系统本身必须是可应急的，对于所依赖的下游服务、存储，应具备灾备切换或功能降级的能力。对于每一个功能模块，应设计动态开关。在紧急情况下，可以动态关闭部分功能模块，容忍部分业务平滑降级的情况下保证整体服务基本可用。

应急预案：建立应急平台，应急预案、应急开关系统化、集中化管理，专人负责。

定期演练：所有应急预案应定期演练保鲜，无损预案可以高频次、常态化演练；有损预案在可控条件下谨慎演练，如夜间业务低峰期。

应急度量：预案效果需要有可量化的度量指标，如应急效果、时长等，方便在复盘阶段提供有效抓手，持续优化预案。

从手动应急到自动应急：前期以人工盯屏监控、人工决策、人工执行；后期部分成熟固化的场景，可以结合监控数据做自动决策、自动执行，实现故障快速自愈。



来源：公开资料整理

图 17 应急平台能力框架图

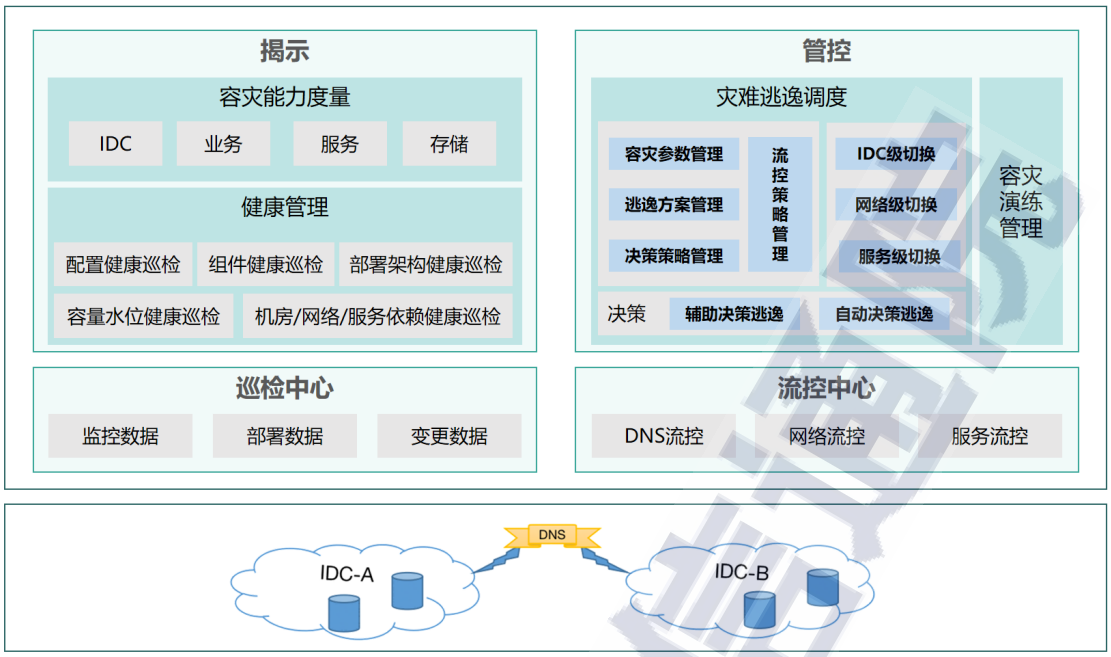
（2） 容灾管理

分布式系统的扩展伸缩架构能力天然为灾难逃逸提供了前置条件，多数企业也会建立多机房冗余的灾难逃逸能力，并且也会通过周期性的容灾演练验证保鲜。由于缺少平台化的约束及保障机制，通常一些无意识的配置、依赖、部署架构等变化会影响已有的容灾设计导致整体容灾能力的衰退，在此情况下，形成统一的容灾管理平台化能力就显得意义非凡。我们建议在企业发展的合适阶段，可以逐步建立起平台化的容灾管理能力，驱动逐步实现：IDC、网络、服务多级灾难逃逸能力，以达到更从容的灾难应对能力。

容灾管理主要分位容灾揭示、容灾管控两部分，其中巡检中心和流控中心作为容灾揭示和容灾管控的基础工具依赖。

容灾揭示：主要从机房、业务、服务、存储 4 个层面进行度量，通过一系列的健康巡检给出预示和告警。

容灾管控: 容灾的主要手段是进行流量调度，确保业务的连续性，为了支撑管控会进行相关的参数管理、方案及策略管理。其决策手段可以根据方案的成熟度进行辅助决策或自动决策。同时，为了确保管控方案的可行性，需要定期进行容灾演练的管理。图 18 示意了容灾管理能力建设思路。



来源：公开资料整理

图 18 容灾管理能力建设框架图

应用多活作为容灾技术的一种高级形态，通过在同城或异地机房建立一套与本地生产系统部分或全部对应的生产系统，所有机房内的应用同时对外提供服务。当灾难发生时，多活系统可在分钟级内实现业务流量切换。应用多活架构主要由以下几方面能力构成：

管控层：是多活的管理控制层，通过多活控制台、应用接入配置管理、日常演练、多活切换、管控高可用等关键动作，实现对应用多活的管理控制，保障多活管理的高效性和高可靠。

接入层：是业务多活的核心入口，是整个多活业务开放的汇聚点，具备对流量接入的统一管理能力，包括流量保护、流量纠错、策略路由等。

应用层：是业务应用流量主经的链路，涵盖应用多活的端到端生命周期管理，包含应用架构设计、消息中间件、微服务中间件等。

数据层：是对整个系统中数据存储相关组件的规范和设计约束，确保整个系统数据层的可靠性、可用性和数据持久化程度满足多活要求。包括业务应用数据读写、数据存储、数据同步等关键动作。

基础设施层：是基于计算，存储，网络等层面的基础设施多活能力建设，确保应用多活底座高可用。

来源：中国信息通信研究院

图 19 应用多活能力框架图

六、分布式系统稳定性建设行业特点

本章关注不同行业在推进分布式稳定性建设过程中的特点，主要从行业特点、稳定性挑战以及解决方法三方面展开介绍，以期为不同行业的分布式系统稳定性能力建设提供有价值的参考。

（一）互联网业

1.互联网行业特点及技术挑战

互联网行业在庞大的线上业务规模下海量的并发请求、敏捷的运营诉求驱动着应用从单体服务向微服务、分布式系统演进。受益于云原生的 DevOps、Kubernetes、微服务、服务网格等技术红利，应用的上线下线、发布变更、容量管理、服务治理等运营效率获得了极大提升。运营效率和用户价值的交付效率得到提升的同时，复杂架构下为系统稳定性保障也带来了新的挑战，主要表现为以下方面:

微服务间调用关系错综复杂，给服务性能瓶颈分析、快速定位影响评估范围和根因分析等方面带来了诸多的挑战。云原生一线开发/运维人员时常会面临“服务调用关系错综复杂，如何快速定位问题根因”、“某服务发生异常，如何快速评估影响范围”以及“如何快速分析复杂系统的服务瓶颈点”等问题。

在复杂的分布式系统中，**无法阻止故障的发生**，且分布式系统日益庞大，很难评估单个故障对整个系统的影响，如何能在这些异常故障被触发之前，如何尽可能多地识别风险，在复杂的分布式架构中亦为稳定性保障一大挑战。

互联网业务服务具备活动发布节奏快、数量多、访问量大且无法精准预估以及逻辑复杂、上下游依赖复杂等特点，需要耗费大量人力梳理调用关系和放大倍数，且**容量评估不准确**对稳定性保障影响较大。

2.互联网行业系统稳定性解决方案

面临以上挑战，为了保障分布式系统下的服务稳定性，可以从以下方面入手:

架构设计方面，我们认为所有的架构都是不完美的，都存在缺陷，因此我们在做业务架构设计时都必须要考虑服务稳定性保障，如负载均衡、多点容灾、集群化服务、数据多活等能力。

从**组织架构和人员构成**上设立 SRE（Site Reliability Engineering, 稳定性工程）团队和 SRE（Site Reliability Engineer,稳定性工程师）角色。且按照 SRE 前置原则，SRE 角色需要提前介入，将运营阶段可能出现的问题或风险提前在架构设计、编码阶段暴露，提前准备好解决方案，甚至规避问题与风险。

建设可观测性能力，即通过采集业务指标、日志、追踪等数据，构建监控告警能力，并且在“事中”环节用于快速分析与定位问题，同时发现复杂系统的瓶颈点。

建设混沌实验平台，通过模拟现网真实故障来验证服务的“韧性”，让故障在测试或预发布环境提前到来，找出系统的弱点，同时验证系统监控告警的有效性。

建设全链路压测能力，通过与可观测性、混沌实验能力的深度整合，实现模拟真实业务环境全链路压测，达到业务上线前的精准资源评估，主动发现潜在性能、版本缺陷等问题。

建立故障应急机制,故障不可避免，技术人员需要不断去提升 MTBF（Mean Time Between Failure，平均故障间隔），降低 MTTR（Mean Time To Repair，故障平均修复时间），包括“事前”实施大量混沌实验、故障预案、建立“On-Call”机制，“事中”采用打造的工具链，快速发现、分析、定位与解决问题以及“事后”组织总结复盘，沉淀案例经验等措施。

建设 AIOps 能力，随着整个互联网业务急剧膨胀，以及服务类型的复杂多样，“基于人为指定规则”的专家系统逐渐变得力不从心。建设 AIOps 能力，可以基于采集到日志、监控信息、链路、指标等数据，通过机器学习的方式来进一步提升异常检测、根因分析、故障诊断、故障预测、故障自愈等运维能力。

（二）银行业

1.银行业特点及技术挑战

银行业的 IT 系统，一旦出现稳定性问题，不单要面对自身的业务损失，还要面对国家相关部门的监管。银行业与传统行业相比，在关键业务场景的核心诉求具有两个鲜明的业务特性:

保证**账务一致性要求**，在各种复杂的金融业务场景下保证客户资金和银行资金准确无误;

海量高并发的交易处理要求，达到每秒钟万笔以上，因此对分布

式数据库的要求更为严格，特别是在高并发、实时性和最终一致性方面。

长期以来，以 IBM 主机为代表的传统集中式架构具有高可靠主机操作系统、成熟的主机中间件业务软件和强大的配套数据库软件等特征。随着分布式转型的变革，如何在基于 X86 平台的分布式体系下，既能满足各行业业务线上化、多样化、高增长，又能保持主机同等的稳定性,成为一个亟待解决的难题。

2.银行业系统稳定性解决方案

针对以上银行业业务特点、当前分布式系统技术现状及稳定性挑战，可以通过以下几个技术点推动银行 IT 系统稳定性提升。

两地三中心实现双活数据中心。针对可用性和稳定性要求高的系统，可建设两个或者多个数据中心，主数据中心承担用户核心业务，其他数据中心承担非核心业务并备份主数据中心的数据和配置。主数据中心整体发生灾难时，备份数据中心可以快速恢复数据和应用，减轻灾难给用户和企业带来的损失。

多层级的故障保护机制提供可靠的基础设施架构。开放平台硬件存在一定的故障率是客观事实，基于云计算技术架构来降低故障影响是切实可行的途径。通过云管平台、资源调度平台的两级调度策略，确保同一业务的不同实例均衡分发到多地多中心的不同故障域，确保单个节点、单个集群甚至单个数据中心发生故障时，不会影响到整体业务的可用性。

云平台故障自愈提升业务连续性。容器化部署的应用具备快速启

动和销毁的特点，结合 Kubernetes 的健康检查机制，可以实现多层次的故障自愈能力。

业务可感知的优雅启停机制实现无损升级。借助分布式平台提供的精细化、业务可感知的优雅启停策略，支持研发人员对系统预热进度的自动化探测和状态自适应改变，解决规模化部署后业务系统升级出现的交易波动和交易瞬时问题，提高系统的业务连续性。

（三）证券业

1.证券业特点及技术挑战

证券行业的高并发业务特点集中在开市的四个小时内，业务停滞的每一秒都可能带来巨大损失，因此其对于业务连续性的诉求极高。

随着近十年的市场发展，整个市场交易量也由千亿级发展到万亿级，原来传统的集中式交易系统的弊端愈发突出，IT 架构向分布式转型成为必然趋势，分布式架构拥有的高可用、低延时、高并发、灵活扩展的相比集中式交易系统的优势明显。分布式架构提升了系统的业务承载能力，但同时也让整个系统变得更加复杂，运维的难度在随之提升。运维的过程难点主要体现在以下方面：

市场业务波动的不确定性，交易业务量的激增的随机性相比一些其他行业高，政治、经济、行业、公司等因素都能影响市场，而不像微信春节红包、双 11 等具有明显周期特征；

业务系统繁杂，在微服务的改造过程中会长期保持着老系统，导致新老系统相互交织耦合；

故障预防难，靠传统的测试方法很难有效保障整个系统的稳定性；

故障定位难，系统的分散、数据分散、业务链路长等问题导致故障分析定位难。

2.证券业系统稳定性解决方案

证券行业大部分核心业务主要集中在 9:30-11:30，13:00-15:00 这 4 小时，有一定时间窗口对系统进行升级维护，使得行业在生产环境对系统进行稳定性评估提供一个低成本、高回报的良好基础。某证券公司围绕“以业务连续性为宗旨”，从“事前”、“事中”、“事后”的三个维度进行了丰富的探索实践。

首先，“事前”传统的单系统性能容量评估发展为全链路业务性能容量评估，从模拟环境到生产等比例环境的容量评估；其次，“事中”将业务链路上的监控“孤岛”进行统一整合，从基础架构监控、网络性能监控、应用性能监控三个维度构建统一监控平台；“事后”的应急处置的自动化率、智能化率的提升。

发展至今天，行业部分头部券商已引入混沌工程实验对系统稳定性保障手段进行更深层次的强化提升，有效的助力了“事前”、“事中”、“事后”的建设成果。

（四）通信业

1.通信业特点及技术挑战

通信行业主要包括设备商和运营商，运营商在整个产业链条中占据核心地位。传统运营商更多的是承担“管道”的角色，即提供网络、基站等基础设施，管道中的内容和服务鲜有涉猎。随着移动互联网和云计算的发展，管道价值不断下降，各大运营商纷纷探索转型之路。

经过多年发展，运营商有了足够多的用户信息、终端信息等数据，也有一定数量级的传统 IT 系统，这些在转型期既是优势，也是压力。运营商的业务覆盖面广，几乎贯穿于每个人的日常生活中，一旦出现问题，对用户体验和品牌形象都将造成重大损失，系统的稳定性保障面临巨大挑战。

新旧系统的共存和过渡。云计算带来更大的数据处理能力，但是从传统系统迁移至云不是一簇而就的，相当长时间内会存在多种系统的共存，如何做好数据一致性、系统平滑割接是重要命题。

和互联网公司越来越接近的软件产品。以基站为代表的网络设施肯定还是运营商的基础，但是随着运营商数智化转型的进程，各种软件产品也被不断推出，云计算、分布式等各种新技术也被广泛应用；如何把稳定性保障嵌入到软件生命周期的各个环节，新技术的人才和资源投入也是需要考虑的问题。

子公司/子系统分散。运营商一般包括数十个省公司和若干子公司，也就对应着很多子系统、很多稳定性保障方案和工具，存在一定程度的资源浪费，如果把各方面的能力整合起来，做到策略、方案、工具一致，不仅节约资源，也能更好的保障系统稳定性，但是实施起来有相当难度。

2.通信业系统稳定性解决方案

针对以上特点或痛点，运营商转型期保障系统稳定性的方法如下：

有序推进各子系统迁移至云，充分测试后适时割接，保证用户无感知；

实施混沌工程，混沌工程适用于大规模的随机故障，非常匹配运营商系统规模宏大、架构复杂的特点；

把稳定性保障嵌入到软件产品的整个生命周期，从设计到开发、测试、上线和监控，都匹配合适的稳定性保障手段。

（五）云服务业

1.云服务业特点及技术挑战

云计算系统作为现代复杂的分布式系统的典型代表之一，其稳定性表现至关重要。相比较其他行业系统，云计算系统稳定性的特征体现在以下方面。

客户业务、技术形态多元。云计算系统对人工智能、大数据、区块链等创新技术的不断整合，服务各行业细分领域形成的多元形态，满足企业混合云/分布式云诉求的架构更新，为实现稳定性设计增加复杂度。

云服务提供商将稳定性建设贯穿于云平台初始规划、设计实现和运维管理的方方面面。对于云计算系统来说，存在多种影响稳定性的因素，主要归纳为硬件故障、软件异常和人为操作等类型。作为云计算系统的设计建设者，云服务提供商运用测试、代码审查和语法扫描等软件工程技术，结合声明式设计、分布式追踪和混沌工程等云原生实践，同时规范生产环境的变更操作制度和过程，保障云计算系统的运行稳定。

2.云服务业稳定性解决方案

实现云计算系统稳定性的途径主要包括：

容量计划。云计算系统的容量计划建立在资产管理和配置管理的基础上，为云计算系统运行环境提供云资源在数量、规格和位置等方面的规划和跟踪。实现容量计划的过程包括建立资源模型、监控分析资源使用、调优资源交付和持续跟踪资源水位等步骤。

稳定性设计。运用云原生技术和实践强化云计算系统的稳定性设计。针对云计算系统，使用分布式追踪技术洞察系统 API 调用情况，进行事务监控和服务依赖分析，辅助故障根因定位；实施混沌工程，检验系统面对各种错误的反应，建立系统承受生产环境业务压力的信心。

过程保障。制定云计算系统的版本发布和线上变更规范，建立生产环境运维流程制度，控制人为误操作对系统造成的影响范围。

（六）零售业

1.零售业特点及技术挑战

零售行业业务特点在于线上服务和线下服务同时进行，互相依赖，系统高可用尤为重要，当系统产生异常，线上用户无法进行下单，线下服务的门店无法给用户进行结账，会对用户体验带来巨大且无法挽回的影响。

零售行业系统迭代速度快，系统每周上线进行更新，系统稳定性保障非常重要，各个系统的保障措施、应急方案、回滚方案必须做到非常完善，同时有相应的团队进行技术保障支持，确保系统线上运行稳定。

近年来传统电商行业进行的大促培养了固定的大促模式和用户

群体，大促日期也相对固定，比如 618、11.11、12.12 等，用户群体更偏向于年轻群体,零售行业区别于传统电商行业,大促时间在 618、11.11、12.12 的基础上，同时包括了中国传统的节假日，业务峰值周末峰值高于平时峰值，这点区别于传统电商行业，用户群体分布在各个年龄段，线下门店的系统更要做到好用、便捷，给百姓带来良好的购物体验。

2.零售业系统稳定性解决方案

零售行业系统稳定保障一般从如下入手:

系统链路的梳理，分析薄弱点和瓶颈点，制定相应的应急预案，做到对系统了如指掌，应急方案做到有效、高效，而不是纸上谈兵;

系统监控全面且可视化，包括应用监控、基础中间件监控、业务监控等，同时能够明确高效通知系统负责人，监控报警做到聚合，防止报警轰炸;

系统高可用，具备容灾、异地多活能力，机房间切换无感知，对用户正常访问的影响降到最低;

系统稳定性建设，依赖服务的降级方案制定，核心业务和非核心业务解耦，数据读写分离，系统间调用阈值制定，系统自愈、熔断降级方案制定，数据慢查询治理;

故障演练，制定故障演练方案和编排演练场景，模拟应用、网络、依赖服务、硬件、中间件、数据库等故障，考核系统健壮性以及故障恢复时间，同时评估应急方案有效性;

全链路压测，通过对核心系统（首页、商详页、推荐搜索、购物

车、结算页、订单、物流履约等）进行性能评估，考核系统可靠性以及最大处理能力，确保大促过程中系统稳定，能够应对突发流量对系统带来的冲击。

（七）能源业

1.能源业特点及技术挑战

能源企业数字化系统对安全稳定运行具有很高的要求，根据数字化应用不同的等级，呈现出不同的要求。其中，生产控制类系统系统安全稳定运行要求最高，且受国家能源局监督，系统不稳定事件会影响能源供应的安全，更会影响到国民生计及社会秩序，稳定性要求最高。其次，是面向企业客户类系统，如电费缴纳，用户报装及迁改等，系统不稳定不仅影响公司营业收入，也可能引发社会舆论影响，对稳定性要求较高。

2.能源业系统稳定性解决方案

近年来能源企业深度应用云计算新技术，遵循全生命周期管控，全技术栈支持，全方位运行保障的原则，保障业务应用的连续可靠运行。

全生命周期管控方面，制定了云上应用系统高可靠性相关配置要求，指引应用系统规范上云，在应用系统可研阶段、需求设计开发测试、部署实施等各阶段进行技术管控，保证可靠性各项要求切实落地。

在全技术栈支持方面，基础设施层实现资源冗余，平台软件与应用技术互相配合，达到故障隔离与故障自愈的目标，在应用顶层架构方面采用同城双活、应用级灾备和数据级灾备支持跨域切换，保证系

统可以持续提供服务。

在全方位运行保障方面，实现运行保障与应急演练互相促进的方式，通过实战演练，把系统“薄弱”环节纳入检查因子，通过实际注入故障等方式，做到提前发现问题，提前解决问题，提前检验应急保障的实战能力。行业部分典型企业通过引入混沌工程，优化全生命周期管控，全技术栈支持和全方位运行保障的效率，提升能源企业数字化应用的建设成效。

七、分布式系统稳定性建设展望

（一）人才、生态、标准亟待关注，多重措施提升稳定性发展水平

1.专业化人才稀缺，急需拓宽技术交流平台

保障企业 IT 系统稳定高效地持续运行涉及的技术面广、难度大，是一项技术门槛较高的工作，很多企业缺乏相关专业技术人才及技术积累，市场急需专业技术人才，当前需要更多地借助供应商的知识、技术、人力来完成相应能力建设，国内 IT 系统稳定性保障服务领域蕴藏巨大商机。目前系统稳定性保障赛道足够宽，整体而言参与者并不多，行业发展尚处于起步阶段，头部厂商技术一枝独秀，中部厂商能力各有千秋。同时资本市场已经开始关注、布局这个新兴赛道的准独角兽企业。

重视专业技术人才培养，搭建专业技术交流平台，推动专业人才成长和专业能力沉淀。针对市场专业人才稀缺，行业发展仍处于初级阶段的情况，需要培养人才的专业化、专门化、精细化能力，营造良好的技术交流氛围，提升各供应商能动性与创造性，促进系统稳定性保障行业高速发展。在聚焦能力短板，强化专业人才培养方面，混沌工程实验室将持续挖掘系统稳定性保障领域技术痛点，面向预研人员、开发者和实验室成员不定期推出一系列前沿技术课程。在拓宽技术交流渠道、为行业中坚技术力量发声方面，混沌工程实验室为专业技术人才提供跨企业的沟通渠道，开展面向不同行业、背景人才的专题交流活动。

2.稳定性生态协同低效，急需赋能产业协同创新

系统稳定性保障热度攀升，同时系统稳定性领域涉及面广且深，单个企业难以提供覆盖市场所有需求的产品，需要协同行业内多家企业合作，各取所长，促成丰富、完善的稳定性保障生态。目前，由于稳定性保障赛道尚处于起步阶段，供应商之间存在争夺市场资源、规避技术风险等方面的考量，企业各自为营，导致产品趋于同质化，不利于稳定性产业繁荣。推广生态开放的价值、构建稳定性保障服务矩阵，将有利于产业发展，也是混沌工程实验室未来的工作方向。

产学研多方共同打造产业开放生态，驱使产业高质量发展，协同推进产业创新升级。面对产业发展待协调，产品同质化严重，创新不足的现状，应联合产学研多方协同打造开放共享的产业生态，博采众长，推动系统稳定性保障技术与产品创新，产业共同进步。打造全方位创新资源共享平台，提供技术咨询、技术对接、产学研合作渠道等一体化服务，为系统稳定性保障产业的发展提供动能。混沌工程实验室将作为企业之间沟通、交流、合作的桥梁，携手共建“协同”平台、共享“创新”资源、共创“生态”体系、共同“赋能”未来。

3.标准体系研制滞后，规圆矩方行稳致远

稳定性领域的技术点丰富，相关标准体系在持续完善中，但标准研制进度不及用户需求的爆发式增长及相关技术/产品的能力迭代。标准的缺乏，致使稳定性相关产品提供方缺乏能力建设指导，产品能力参差不齐；对需求方来讲，缺少产品选型原则和依据，提升需求方 POC 难度。

重视行业标准研究、建设工作，围绕系统稳定性保障相关技术完善标准体系。不断完善的行业标准体系建设，是促进系统稳定性保障产业高质量发展的强有力支持。基于现阶段系统稳定性保障领域标准体系建设的主要任务和方向，中国信通院积极发挥标准技术研究优势，紧随前沿技术发展趋势，以满足产业需求为导向，多次召开标准研讨会，针对系统稳定性保障领域的各个专题展开深入讨论，并与行业专家深度合作，共同完成系统稳定性保障系列标准的编写工作。

（二）顺应时代发展需求，推动稳定性建设进入新阶段

1.稳定性建设能力发展不均，传统行业需求蓄势待发

系统稳定性是企业对外提供服务的基础，但稳定性保障技术**在不同行业的应用程度不均衡且不充分**。以混沌工程为例，互联网公司中混沌工程的应用已经比较深入，从基础资源、中间件、微服务扩展到上层故障演练、容灾多活、攻防活动等。相对而言，国内相当多的传统企业还停留在分布式系统改造升级后的基础资源验证、业务场景探索、混沌价值不明阶段。而在数字化转型浪潮下，传统企业开始迈向数字化与智能化，其稳定性保障和建设需求也随之高涨，正逐步丰富系统稳定性建设赛道的商机。

2.企业架构阻碍稳定性建设，组织观念正逐步进化

面对系统分布式化带来的一系列稳定性问题，需要组织架构上的配合调整以及思想观念的变革，**牵动团队全员共同建设稳定性保障组织**。稳定性保障组织建设主要体现在两个方面：一是**组建 SRE 团队**，横跨各业务线调动全公司资源（包括但不限于技术、法务、合规），

确保应对重点项目的稳定性建设时，能从各方面给予保障支持。二是**构建 DevOps 流程机制**，统一软件开发（Dev）和软件运维（Ops），让运维人员介入到开发过程中，了解开发人员使用的系统架构和技术路线，从而制定适当的运维方案，同时让开发人员在运维初期参与到系统部署中，并提供系统部署的优化建议。

3.过度依赖开源致“懒”，倡导创新采纳开源技术

国内外有众多优秀的稳定性领域开源工具供企业、个人用户体验使用，如表 6 所示，企业和个人用户应拥抱创新开放的开源，推进开源协作模式在行业中的应用，这样可以避免分散重复开发，赋能中小企业低成本发展，但使用开源框架时，要积极创新、勇于创新，**避免“套路化”依赖**已有开源工具，强调创新而有效地开源，提高对开源技术的应用水平和自主可控能力。

表 6 中美稳定性工具开源情况

稳定性工具	中国	美国
混沌工程工具	ChaosBlade ChaosMesh OpenChaos	Chaos Toolkit Chaos Monkey AWSSSMChaosRunner Chaos Gamedays
压测工具	XPocket	Vegeta loadUI Apache JMeter Locust
可观测性工具	Kindling DataKit	OpenTracing OpenCensus OpenTelemetry

		Zipkin

来源：公开资料整理

CAICT 中国信通院

附录 1

表 7 稳定性守护者列表[1]

稳定性能力域	守护者名称
混沌工程能力	阿里云计算有限公司
	华为云计算技术有限公司
	中国移动通信集团有限公司信息技术中心
	平凯星辰（北京）科技有限公司
	南京争锋信息科技有限公司
	建信金融科技有限责任公司
	北京同创永益科技发展有限公司
	蚂蚁科技集团股份有限公司
	杭州笨马网络技术有限公司
	深圳市腾讯计算机系统有限公司
应用多活能力	阿里云计算有限公司
	建信金融科技有限责任公司
全链路压测能力	阿里云计算有限公司
	蚂蚁科技集团股份有限公司
	杭州笨马网络技术有限公司
可观测性能力	阿里云计算有限公司
	蚂蚁科技集团股份有限公司
	上海驻云信息科技有限公司
	北京优特捷信息技术有限公司
	中电云数智科技有限公司
	优维科技（深圳）有限公司
	北京基调网络股份有限公司

来源：中国信息通信研究院

[1] 截止 2022 年 6 月

附录 2

表 8 混沌工程实验室成员列表[2]

序号	企业名称	类型
1	中国信息通信研究院	理事长单位
2	华泰证券股份有限公司	副理事长单位
3	阿里云计算有限公司	副理事长单位
4	中国工商银行软件开发中心	副理事长单位
5	天翼云科技有限公司	副理事长单位
6	中国移动通信集团有限公司信息技术中心	副理事长单位
7	北京百度网讯科技有限公司	副理事长单位
8	深圳市腾讯计算机系统有限公司	副理事长单位
9	南京争锋信息科技有限公司	副理事长单位
10	华为云计算技术有限公司	副理事长单位
11	平凯星辰（北京）科技有限公司	副理事长单位
12	中国农业银行信息中心	副理事长单位
13	北京银行股份有限公司	副理事长单位
14	杭州笨马网络技术有限公司	副理事长单位
15	北京同创永益科技发展有限公司	副理事长单位
16	蚂蚁科技集团股份有限公司	副理事长单位
17	上海浦东发展银行股份有限公司	副理事长单位
18	建信金融科技有限责任公司	副理事长单位
19	京东科技信息技术有限公司	副理事长单位
20	中信银行股份有限公司软件开发中心	副理事长单位
21	浩鲸云计算科技股份有限公司	理事单位
22	北京火山引擎科技有限公司	理事单位
23	中移（苏州）软件技术有限公司	理事单位
24	南方电网数字电网研究院有限公司	理事单位

[2] 截止 2022 年 6 月

25	阳光保险集团股份有限公司	理事单位
26	四川省农村信用社联合社	理事单位
27	中电金信软件有限公司	理事单位
28	中移（杭州）信息技术有限公司	理事单位
29	北京永辉科技有限公司	理事单位
30	思特沃克软件技术（北京）有限公司	理事单位
31	中兴通讯股份有限公司	理事单位
32	北银金融科技有限责任公司	理事单位
33	中国光大银行股份有限公司	理事单位
34	北京必示科技有限公司	理事单位
35	中信建投证券股份有限公司	理事单位
36	中电云数智科技有限公司	理事单位
37	中国科学院计算技术研究所	理事单位
38	招商银行总行信息技术部	理事单位
39	中关村智联联盟	理事单位
40	中原银行股份有限公司	理事单位
41	上汽通用汽车有限公司	理事单位
42	安信证券股份有限公司	理事单位
43	中泰证券股份有限公司	理事单位
44	上海钧正网络科技有限公司（哈啰出行）	理事单位
45	中国银行股份有限公司	理事单位
46	恒丰银行股份有限公司	理事单位
47	中科南京信息高铁研究院	理事单位
48	申万宏源证券有限公司	成员单位
49	北京水木羽林科技有限公司	成员单位
50	中国联合网络通信有限公司软件研究院	成员单位
51	浙江菜鸟供应链管理有限公司	成员单位
52	东方证券股份有限公司	成员单位

53	太平洋财产保险股份有限公司	成员单位
54	亚信科技(中国）有限公司	成员单位
55	极狐创新（北京）信息技术有限公司	成员单位
56	天翼电子商务有限公司	成员单位
57	浪潮软件集团有限公司	成员单位
58	中国移动通信集团湖南有限公司	成员单位
59	上海富麦信息科技有限公司	成员单位
60	东软集团股份有限公司	成员单位
61	恒生电子股份有限公司	成员单位
62	兴业数字金融服务有限公司	成员单位
63	上交所技术有限公司	成员单位
64	上海银行股份有限公司	成员单位
65	钉钉（中国）信息技术有限公司	成员单位
66	中债金科信息技术有限公司	成员单位
67	杭州微智测信息技术服务有限公司	成员单位
68	济南浪潮数据技术有限公司	成员单位
69	科来网络技术股份有限公司	成员单位
70	上海有孚网络股份有限公司	成员单位
71	江苏苏宁银行股份有限公司	成员单位
72	优维科技(深圳)有限公司	成员单位

来源：中国信息通信研究院

中国信息通信研究院 云计算与大数据研究所

地址：北京市海淀区花园北路 52 号

邮编：100191

电话：010-62309514

传真：010-62309514

网址：www.caict.ac.cn