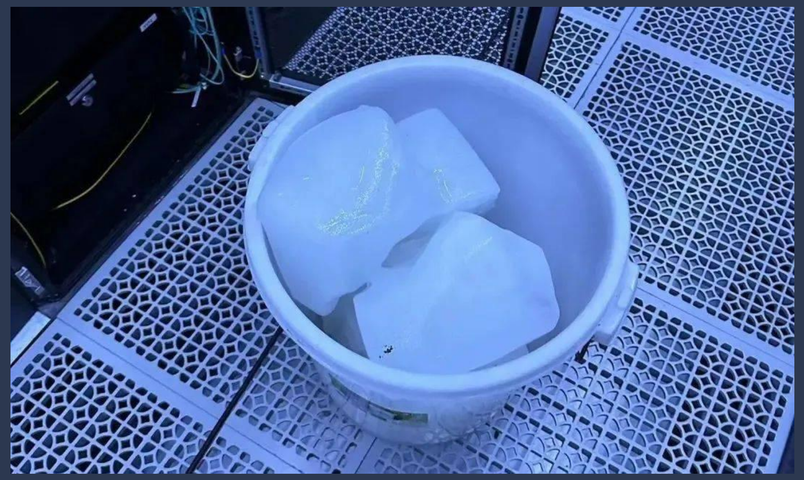
系统稳定与信息安全体系建设与实战经验

刘向阳
美的集团首席信息安全官（CISO）兼软件工程院院长
欧洲科学院院士
IEEE Fellow、IET Fellow、AAIA Fellow
ACM Distinguished Scientist

CIO最怕什么：稳定爆雷、安全爆雷



国际船舶制造巨头宾士域集团因网络攻击损失超6.1亿元

制造业 · 安全内参 · 2023-08-03

发生网络攻击后，宾士域集团被迫暂停部分地区运营，公司花费9天时间才恢复正常运营，造成了巨大的时间损失，严重影响了当季财报。


墨菲定律
你越不想发生的事，往往越会发生

稳定性度量

MTTF (Mean Time To Failure)：平均无故障时间
MTTR (Mean Time To Repair)：平均修复时间

$$Availability = \frac{MTTF}{MTTF + MTTR}$$

Level of Availability	Percent of Uptime	Downtime per Year	Downtime per Day
1 Nine	90%	36.5 days	2.4 hrs.
2 Nines	99%	3.65 days	14 min.
3 Nines	99.9%	8.76 hrs.	86 sec.
4 Nines	99.99%	52.6 min.	8.6 sec.
5 Nines	99.999%	5.25 min.	.86 sec.
6 Nines	99.9999%	31.5 sec.	8.6 msec

容灾系统度量指标：

RTO（Recovery Time Objective）：业务系统恢复运行所需时间的目标

RPO（Recovery Point Objective）：业务数据丢失的最大时间窗口

故障=风险 + 触发

必然性

↓

预防故障

偶然性

↓

减少影响

稳定性建设十板斧

预防故障

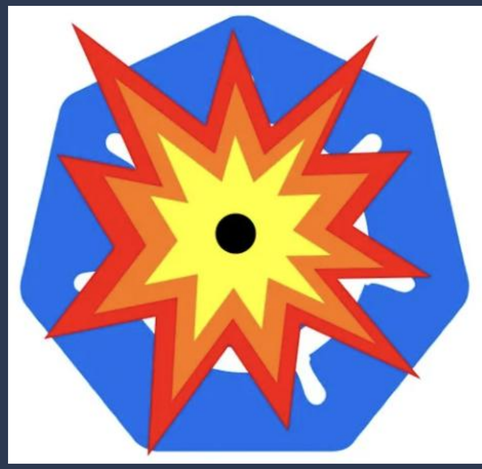
- 架构梳理
- 风险矩阵
- 变更预案
- 日常巡检
- 代码防御

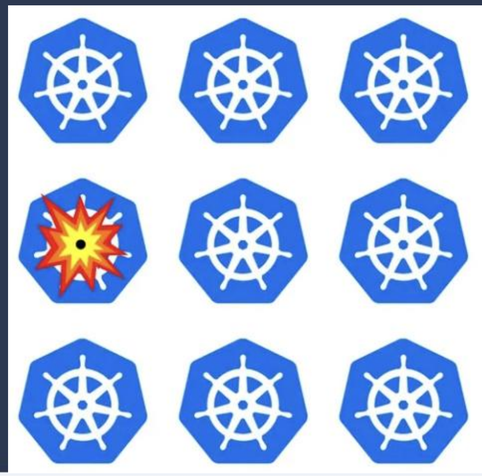
减少影响

- 全面监控
- 应急预案
- 一键逃生
- 故障演练
- 管理制度

系统稳定性建设十板斧之一：架构梳理

- 面向失败的架构设计
  - 高可用：部署冗余架构，消除单点风险
    - 单点消除：硬件单点、存储单点、网络单点、机房单点、依赖单点、注册单点
    - 冗余设计：数据冗余（数据多份副本，DB 一主多备），计算冗余，网络冗余，存储冗余
    - 弱化依赖： 避免强依赖（依赖服务挂掉会导致自己服务挂掉）
    - 分库分表：避免热点
    - 流量管理：限流、降级、熔断
    - 容量管理：动态扩缩容
    - 链路分析：同步调用链路分析，系统拆分粒度，同步调用vs.异步调用
  - 高性能：数据尽量索引，网络带宽延迟CDN，冷热数据分离
  - 高质量：纵向技术分层，横向业务分区
- 灾备设计
  - 冷备、热备
  - 数据级别容灾、应用级容灾、业务级容灾
- 异地多活：两地三中心，三地五中心





系统稳定性建设十板斧之一：架构梳理

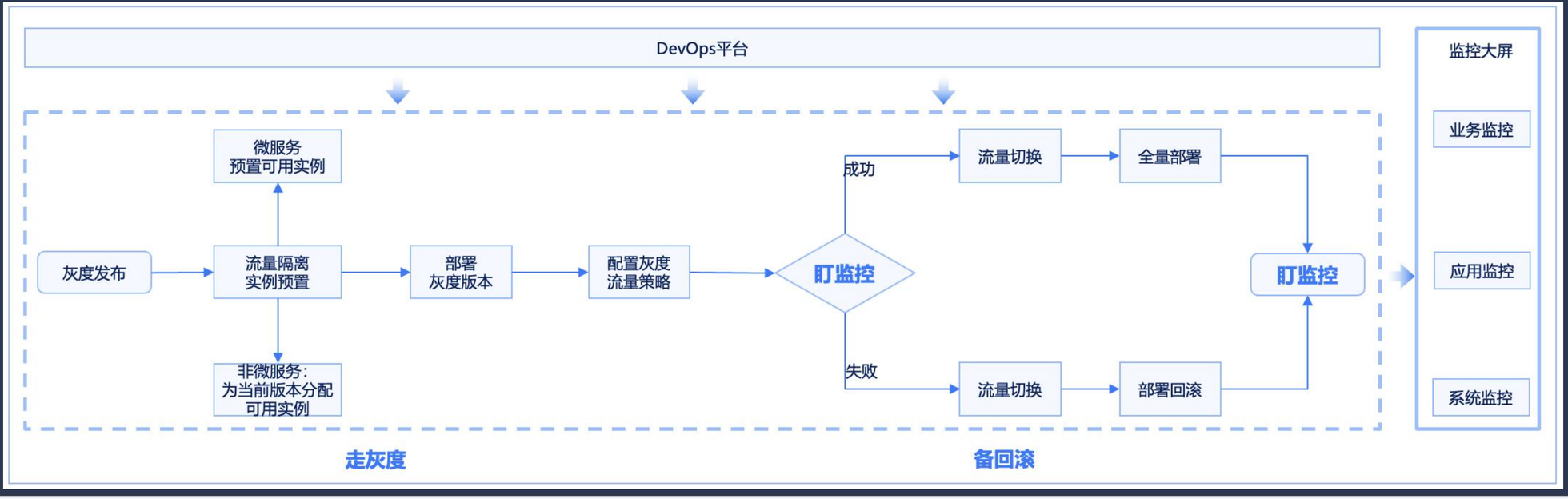
- 软硬件系统的老旧是造成故障的常见根因
  - 用机械盘存索引的对象存储集群
  - 老旧版本的k8s容器集群
- 技术升级
  - 软件升级：坚决避免原地升级
  - 硬件升级：做好预案随时回滚

系统稳定性建设十板斧之二：风险矩阵

- 死法理全：直接死法有限，树状逐层分析
- 风险矩阵模版：
  - 风险信息
  - 故障预防
  - 故障发现
  - 应急预案

系统稳定性建设十板斧之三：变更预案

- 变更种类：软件变更，配置变更，数据库变更，硬件变更，主机变更，网络变更
- 变更原则：走灰度，备回滚，盯监控
- 变更流程：变更方案评审、变更效果验证、业务检查清单
- 变更影响面，联合上下游
- 团队提前做好所有可能变更的预案



系统稳定性建设十板斧之四：日常巡检

- 传统的航空、电力、汽车行业都有巡检机制，保障设备系统正常运转，软件系统也同样需要巡检机制保障业务健康发展，如果优化不及时就很可能导致故障。
- 巡检内容
  - 基础信息：CPU利用率、磁盘利用率、内存利用率、服务器时间同步、日常数据备份文件检查==》关注趋势变化
  - 集群状态：证书过期检查、API通信是否正常、各名称空间下的pod运行状态、ceph共享存储是否正常
  - 业务状态：业务平台登录是否正常、kafka是否积压、kafka消费速率
- 人工巡检==》自动巡检 ==》巡检平台

系统稳定性建设十板斧之五：代码防御

- 防御性编程（Defensive programming）：不仅自己的代码不要有bug，还要防止别人的代码有bug；对每一个软件模块，不仅这个模块不要有bug，还要防止上游的模块有bug。
- 输入检查：检查所有来自系统外部的数据，根据业务逻辑进行检查
  - 在Domain Driven Design中Specification Pattern:用于对对象的属性进行检查，从而保证对象的逻辑正确性。
  - 安全检查，如SQL注入，XSS攻击等
- 异常处理：通过异常处理Java try-catch，处理所有可能的错误，包括系统错误和用户错误
- 自愈代码：在coding过程中，在任何关键环节都要具备稳定性意识。数据校验：实时校验-实时防止bug发作；离线校验：发现脏数据
- 其它问题：日志打印（日志规范要统一）、幂等问题、资源泄露、事务问题
- 自动化测试：流量回放，Fuzzing平台

系统稳定性建设十板斧之六：全面监控

- 系统监控
  - 主机监控：cpu使用率， disk使用率， SWAP使用率 ， mem使用率， 磁盘剩余空间等等
  - 网络监控： ICMP loss，ICMP response time，High bandwidth usage，端口带宽使用率，设备端口状态，设备端口错误包等等
  - 中间件监控：ES集群服务器cpu使用率、GC次数、内存使用率， kafka/rocketmq消费积压数量、 消费积压数量等等
  - 数据库监控： mysql 活跃线程数、最近2分钟QPS、最近5分钟慢sql数量、主从延迟时间、可用连接数、主机swap使用率等等
- 应用监控
  - 异常总数，FullGC次数，慢调用数量，响应时间，请求数， Slowest 5%， Slowest 15%，调用外部接口成功率、响应时间
  - 五慢监控：慢查询，慢消息，慢响应，慢调用，慢缓存
  - 全链路监控：跟踪跨分布式应用的消息，定位失败点和瓶颈；字节码增强技术
- 业务监控
  - 方法：（1）业务用服务调用提供业务指标，监控平台拉取；（2）业务把指标写入日志，监控平台读取日志。
  - 拨测监控：拨测成功率
- 终端监控
  - 页面性能、API性能、多维分析、分阶段测速、卡顿、崩溃分析、特征分析、用户行为分析

系统稳定性建设十板斧之七：应急预案

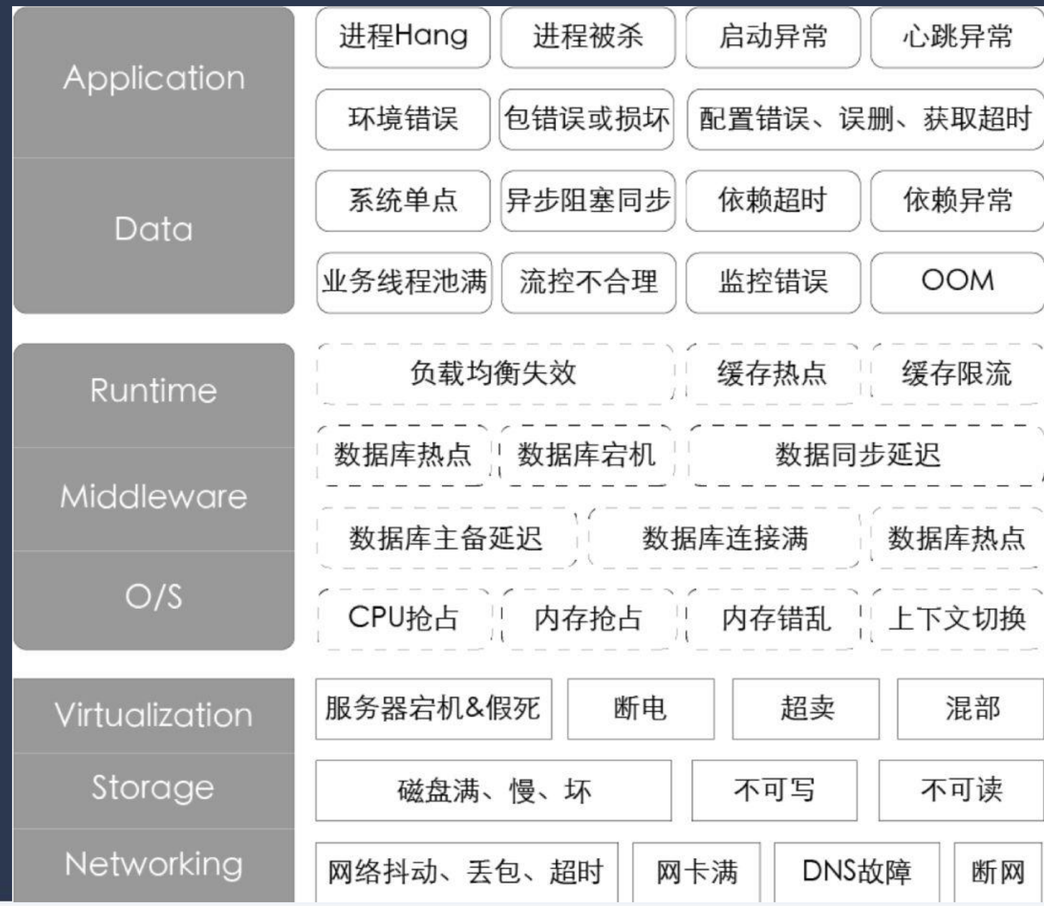
- 每个团队对每一个可能故障，必须制定完善的应急止血流程图
  - 在发生故障的时候，大多数人的脑子都是一片空白。应急预案必须做到“无脑化”。
  - 对各种问题设计止损、兜底、降级开关等策略
  - 人：谁通知，通知谁，谁协调，协调啥，谁决策，决策啥，止血原则等
  - 应急团队：必须事先针对每个可能的故障定义好，对应的群最好事先已经拉好
- 应急原则
  - 预案首要原则是快速止血，即快速恢复业务，不是彻底查找根因
  - 当前应急负责人若短时间内无法解决问题，必须升级处理
  - 应急过程中在不影响用户体验前提下，要保留部分现场和数据，便于恢复后定位分析原因
  - 有明显的资金损失时，要在第一时间升级，快速止损，该条用在金融领域尤为关键

系统稳定性建设十板斧之八：一键逃生

- 对阻断性安全设备，必须提前准备好一键逃生方案：防火墙、WAF、上网行为管理、SSLO等
  - 主备切换可能失败，这时必须迅速bypass，事先做好方案，做好演练

系统稳定性建设十板斧之九：故障演练

- 演练种类：故障演练，灾难演练
- 演练原则：严控范围风险，确保不出故障
- 演练预案：事前准备详细演练预案
- 演练目标：检验预案，锻炼队伍，磨合团队
- 混沌工程：故障注入，控制爆炸半径
- 全链路压测
  - 场景编排、压力模型、数据准备、压力机准备
  - 施压策略、发压调度、压测监控、压测熔断、数据清理
  - 生产压测：流量透传、数据隔离、服务挡板、风险保护
  - 压测报告、性能建议



系统稳定性建设十板斧之十：管理制度

- 值班机制：运维人员轮流值班，7*24小时接收紧急报警， 值班人员必须做到故障第一发现人。所有人员7*24小时电脑不离
- 故障复盘
  - 复盘自己故障：不论是否为定级故障，不论问题大小；每个用户反馈要重视，定位到根本原因。
  - 复盘别人故障：其它团队的故障，其它公司的故障
- 风险矩阵核查：这个故障是否在风险矩阵里面？如果不在，为什么遗漏，举一反三，还有哪些风险被遗漏？如果在，那为什么所有防线被击穿？举一反三，还有哪些风险有类似可以被击穿的防线？怎么做才能避免再次发生这样的故障？怎么做才能是下次即使发生这样的故障，损失能降到最低？
- 监控检讨：谁首先发现的问题？是客户还是自己值班人员？监控是否报警？报警是否被及时处理？
- 应急检讨
  - 应急预案是否有这个故障的预案？预案是否需要修缮？谁写的？其它预案是否需要重新评审？
- 举一反三
  - 每个故障都暴露我们的弱点，举一反三，这个弱点是否在其它地方存在？挖出来，解决掉！
  - 例如：如果是代码问题，那要清查所有其它程序、其它应用的代码，看看类似问题是否存在？

信息安全建设五板斧

信息安全五板斧之一：进不来

- 从外到内
  - 防护DMZ：防火墙+WAF(+SSLO)+API+APT
  - 防护内网：VPN二次认证+远程设备准入+基于身份的访问权限最小化+业务隔离
  - 认证：4A/C4A系统
  - 应用安全
    - DMZ：内部应用迁入内网
    - 外网应用二次认证
    - DevSecOps：SCA，SAST，DAST，IAST，RASP，安全SDK
    - HTTPS改造，应用系统漏洞扫描与整改
  - 邮件二次认证+安全网关+防钓鱼
  - 公有云
    - 网络隔离：DMZ+内网--》公有云应用改造
    - 边界防护：防火墙、WAF
  - 资产管理：CMDB数据校准
- 从内到内
  - 网络隔离：基本全部白名单
  - 生产准入：堡垒机二次认证
  - 办公准入：网络准入（合格终端接入）
- 从内到外
  - 从办公网主动外联：特殊设备白名单
  - 从数据中心主动外联：白名单
- 工控安全
  - 访问严控（白名单）+准入严控+外设严控等
- 弱口令
  - 非口令生成软件生成的口令都是弱口令
    - IM口令+报备除外
    - 口令生成软件

信息安全五板斧之二：能发现

- 服务器（IDC与公有云）：HIDS
- 终端（办公、工控）：EDR+内存安全（指令序列白名单）
- 网络：流量探针、文件沙箱、邮件沙箱、API安全（API调用监控）、APT检测
- 员工：上网行为管理、DLP、严控应用敏感数据导出、大数据访问、导出、权限控制+数字水印
- 大脑：
  - SOC+工控SOC
  - 自研安全数据挖掘系统：UEBA，数据泄露检测等所有SOC难以覆盖的地

信息安全五板斧之三：防泄漏

- 公司电脑：终端DLP，阻止终端层面数据泄漏
- 个人电脑：
  - VPN准入：公司标准化终端才能接入VPN
  - IM系统改造
- 防止拖库：数据泄漏风险监控-基于AI进行数据风险泄漏监控（DRA）
- 数据加密：数据存储加密、传输加密、使用保密（隐私计算），密钥管理系统KMS，密钥全生命周期管理和加解密服务
- UEBA：基于安全大数据分析用户行为管控内部用户风险
- 分类分级：重要数据识别、分类分级、安全管控

信息安全五板斧之四：保合规

- 安全合规体系框架
  - **产品设计**：产品隐私设计指引、隐私协议清晰易懂，需告知用户数据如何被收集/使用/保护，包括授权同意、明确告知、自助注销账号、获取个人信息副本、双清单、申诉机制，收集&传输&共享：明确告知并征得同意、最小化收集和共享。
  - **事前评估**：IT系统个人信息活动记录表、数据保护影响评估报告
  - **安全措施**：数据加密、数据脱敏、访问控制、去标识化、热点事件跟踪、应用上线扫描、API监控、端到端加密。
  - **应急响应**：监管督办、用户投诉、应用市场上架。
  - **持续运营**：法规解读、漏洞应急响应、标准修订、合规外审认证。
  - **等保测评**：物理和环境安全、网络架构安全、边界防护、计算机环境安全、安全管理制度、安全管理岗位、安全体系建设、安全运维管理
- 数据跨境安全合规
  - 个人信息数据出境的目的、范围、方式是否合法合规、非必要不出境。
  - 个人信息非必要出境必须技术整改。
  - 境外数据接收方能否承担保护数据的管理责任和具备保护数据能力评估。
  - 个人信息数据出境是否签署出境合同或其他法律效力文件
  - 本地化数据存储符合各国法规要求。
- APP隐私安全治理重点：
  - 违规、超范围收集、使用个人信息。
  - 不给权限就不给用
  - 缺乏用户主体权利，账号注销难。
  - 未经同意自启动和关联启动。
  - 欺骗误导强迫用户行为。

信息安全五板斧之五：重运营

- 组织：在全美的范围内建设CISO体系，有事及时同步
- 7*24小时应急响应机制
- 安全事件处理原则：及时止血、刨根问底、举一反三
- 平台：SOC平台，自研攻击模拟BAS平台
- 多方联动：信息安全部（方案、培训、指引）+业务方（执行落实）
  - 事业部园区安全（含工控安全）
  - 公有云安全
  - 海外安全
- 培训：员工制度及安全意识提升
- 信息安全从业人员必须具备推土机精神

THANKS

软件正在重新定义世界

Software Is Redefining The World