运维实际工作部分

系统运维

系统运维负责IDC、网络、CDN和基础服务的建设(LVS、NTP、DNS);负责资产管理，服务器选型、交付和维修。主机托管(机位、机架、VIP机房出租)、资源出租(如虚拟主机业务、数据存储服务)、系统维护(系统配置、数据备份、故障排除服务)、管理服务(如带宽管理、流量分析、负载均衡、入侵检测、系统漏洞诊断)，以及其他支撑、运行服务等。

应用运维

应用运维负责线上服务的变更、服务状态监控、服务容灾和数据备份等工作，对服务进行例行排查、故障应急处理等工作。

数据库运维

数据库运维负责数据存储方案设计、数据库表设计、索引设计和SQL优化，对数据库进行变更、监控、备份、高可用设计等工作。

安全运维

运维安全负责网络、系统和业务等方面的安全加固工作，进行常规的安全扫描、渗透测试，进行安全工具和系统研发以及安全事件应急处理。

事件响应/紧急处理

事前有哪些准备工作，整个部分如何快速响应，事后如何应对，哪些必须要做，哪些要做的更好。

运维五大要素

### 可 靠 性

网站必须一直处于开启状态、拥有良好的访问速度和可靠性。就这么简单。你的客户和用户都 是比较忙的，可能会随时访问你的系统，如果你的用户位于不同的地理区域，更是可能在晚上或深 夜访问你的网站。所以，取得经济效益如销售产品、塑造高品质的品牌和获得客户信任，不惜一切 代价使系统持续处于稳定运行状态是头等大事。

#### 您需要考虑：

完全冗余的硬件和服务器 - 发生故障，硬件故障以及软件故障。真正的高可靠性系统，是一个 全冗余系统，从下面的IDC, 到双重IDC 供给，再到防火墙、交换机、物理服务器、虚拟机、负载均 衡器、网络服务器、数据库、缓存以及其它一切东西。为保证系统处于持续运行状态，您就得采用 双冗余架构。 全天候深度监控 - 即使您已经采用全冗余系统架构，您也得积极地避免某些事件发生，或者 至少您必须能够快速侦测并处理此类事件。在事情未发生之前，就对成百上千个数据点进行深度监 控，能够帮助您及时发现问题并快速报告严重问题，以便问题能够尽快得到解决，从而维护系统的 可靠性。 可靠性设计- 对于现今系统而言，必须在设计阶段就考虑到可靠性，从架构到硬件和配置再到 运维和流程，都必须把系统的正常运行时间设计到最长、可靠性设计到最佳。虽然您可以选择多种 简单的设计解决方案，但是这些方案对于维护系统整体可靠性的帮助非常小，所以，您应当采用复 杂的工具如PHP过载探测器、日志分析器、HA Proxy以及Keepalived。

### 性 能

现代网站必须能够快速访问，因为即使增加一秒种的下载时间，也会丢失用户。即使网站访问

速度仅有几秒种的差距，也可能成为业务成败的天壤之别。所以，必须精妙地设计系统，以便在各 个层级上取得最佳的性能，从硬件和网络一直到服务、系统、应用程序、页面结构和[CDN](https://cloud.tencent.com/product/cdn?from=10680" \t "https://cloud.tencent.com/developer/article/_blank)。

#### 您需要考虑：

精妙的设计 - 系统速度快取决于性能工程，或者采用最佳的设计、工具和流程来构建最佳性能 的系统。良好的代码、代码审查、瓶颈分析、现代化的技术、使用有限的功能特点以及常规应用程 序性能工程等，在构建快速系统方面都发挥着重大作用。 应用设计 - 必须在各方面都取得平衡，尤其是在满足现代特点和平台的灵活性以及系统稳定 运行之间必须取得平衡。如今许多工具和框架都非常灵活，可以灵活地进行更改或再利用，但是当 系统扩展后，性能不会很好。采用更多的硬件可以改善这种状况，但是改善的程度也是有限的，所 以，寻找合适的方法，以便终端用户能够快速地访问系统就变得至关重要。 性能监控 - 性能监控是保持良好系统性能的一项关键措施。不仅要在内部系统的服务器和运 维层级上实施监控，而且也要从网络和用户的角度，对系统外部进行监控。运维监控包括：常规的 CPU、RAM、以及磁盘输入/输出监控，还包括对许多与网络服务器、应用程序和代码、数据库等相 关的特定服务问题的监控，重点是对能够提高和/或降低大型系统性能的关键变量进行监控。 系统分析 - 最好的系统会采用像New Relic的性能工具来对代码进行深度监控和分析，以便发现 瓶颈所在以及可以改善的地方。此外，最佳的监控工具能够对系统进行模块化的、整体性的监控和 分析，从基本的代码分析到关键事件的外部调用如数据库以及其它服务，比如搜索、社交媒体以及 外部依赖关系，所有这些都会对系统整体性能和用户体验产生重大影响。

### 可 扩 展 性

性能是很重要的，但是真正成功的站点必需在成名后能够拥有良好的整体性能。许多站点或系 统因为不能够满足不断增长的需求和负荷，不能够符合成功发展的需要而夭折。所以，系统除了必 需具备常规性能特点之外，还必须能够在现有的和将来的硬件/云上发展壮大，必须能够使用现代技 术和技能来扩展系统架构。

#### 您需要考虑：

支持横向和纵向扩展的系统架构 - 扩展工程不同于性能工程，因为对于小型简单系统而言，提 高系统速度是轻而易举的事情，但是要对子系统或平行运维系统/平台进行扩展却并不容易。对于传 统上难以实现系统扩展的区域如数据库性能，实现系统扩展更是一个问题，因为对于数据库性能而 言，读扩展和写扩展必须采用不同的动态的系统架构解决方案。其它扩展问题包括：会话管理、数 据缓存、静态资产共享。 开展深度的系统性能监控以便发现可以改进的区域 - 基本监控是有益于系统运维的，但是还必 需采用更深层次的系统性能监控，以便对系统扩展事宜给与指导。这包括从整体上监控影响操作系 统和硬件性能的一切因素，以及广泛的数据库性能监控。 加载测试 - 每个系统都不一样，在高峰加载时表现出来的性能也各不相同。您会在意想不到的 地方发现瓶颈所在，如果在测试阶段就能够发现这些瓶颈的话，就能够轻易地进行纠正，避免影响 真实用户。虽然，实现合理的加载测试并不是一件容易的事情，但还是要在完美测试以及测试所需 要的时间或资源方面取得平衡（尤其是对于生产系统而言）。 容量规划流程 - 根据加载测试、监控以及扩展性分析，对整体系统能力进行规划，这样有助于 界定系统扩展规模、明确瓶颈所在；并确定高峰流量、促销事件以及常规增长时的动态可用空间。 容量规划还可以构建增长模型，以便把业务目标如用户流量连接到流量模型，这样便可以调试系统 负荷以及响应情况，以便在系统扩展后，能够使系统负荷和响应情况符合实际需求。

### 安 全 性

数据是有价值的，也许，对于许多网站和公司而言是最有价值的，所以，每个人的首要任务就 是保护数据。虽然，每个人都有义务保证安全性，但是，必须将安全设计到系统中才能实现真正的 安全。此外，必须在每个层级上都实现安全保护，因为这就像链条一样，链条的强度取决于最薄弱 的环节，所以，系统安全的薄弱环节很可能存在于技术或代码组中。黑客们是探索系统薄弱环节并 秘密潜入系统的专家，这就要求系统安全专家能够持续保护系统。

#### 您需要考虑：

安全架构和安全代码 - 安全系统源于安全的思维，只有把安全牢记在心，才能构建出设计良好 的系统并编写出安全代码。这些包括采用最佳实践、正确地把事情做好并且一直以受限的权限进行 工作，并且假设一切都有可能受到安全威胁。 最佳实践包括开发 - 正确地进行系统开发对于构建安全系统而言显得至关重要，系统开发必须 遵循严格的流程和架构，比如采用预备好的SQL以防SQL注入攻击。采用先进的工具如静态代码分 析仪，对于保持高质量的代码是非常重要的，采用这些工具之后，甚至无需检测代码。代码审核工 具以及第3方工具在系统开发方面也具有极大的作用。 在各个层次上隔离用户 - 用户隔离对于设置不同的用户权限以及对于用户跟踪/审计而言，都具 有非常重要的意义，对于查证哪个系统/哪个人在哪里做了什么事也具有非常重要的作用。每个人/程 序应当有各自独立的用户名，不得共享用户名，从用户列表和程序列表中应当能够很明确地查看到 系统开机时所发生的一切事项。这种做法也能够帮助程序开发员和运维团队成员强行地把安全观牢 记在心。 频繁的秘密潜入和公开攻击测试 - 只有当系统运维在公共网络并且能够承受专业的黑客攻击人 员的攻击的时候，该系统才称得上是一个好系统。因此，为了尽可能地确保系统安全，最好采用如 下方法进行系统安全测试：黑客秘密潜入测试、扫描测试、审计测试以及频繁的安全审查。 用于实时保护的IPS/安全模块 - 不管采用何等精妙的系统设计和测试，也不可能消除一切薄弱 环节，部分薄弱环节可能会因为第三方工具、应用程序服务器和各种服务的缺陷仍旧留在系统中。 良好的入侵检测系统通常是系统安全的最后一道防线，该系统可以发现非法访问、非法扫描和非法 入侵。虽然这些系统在设置、管理和监控方面都有很大的难度，但是对于关键系统而言，不失为最 佳的一线防御系统。 在各个层次上，使用防火墙和工具进行深度防御 - 只有在各方面、各层次上都是安全的系统才 是真正的安全系统。所谓“深度防御”，是指虽然系统的某一层或某几层被攻破，但是仍旧能够合 理地保护关键资产如数据库。因此，首先要在各个层次上部署防火墙，从面向互联网的公共网络到 内部系统网络再到本机iptables。对每个操作系统以及各项服务都采用良好的设计和其它安全防范原 则，有利于最大限度地维持系统安全性。

### Cost Savings

如果可以使用无限的资源和金钱，那么任何人都可以构建最完美的系统。但事实上，许多公司 是无法这样运维的，并且管控总成本已经成为成功的关键（尤其是对于小型公司而言）。所以，必 须合理安排各项事宜，采用非常经济的架构以及先进的但不必是最前沿的科技、系统和实践，来实 现所有这些关键要素，尽可能以最低的成本，采用上述各种方案，达成所有目标。

#### 您需要考虑：

在各个层次上优化配置 - 在系统构建中，若想省钱是轻而易举的事情，但是若是以关键要素为 代价，如可靠性、性能、可扩展性甚至是安全性，那就不值得了。不合理地节约成本可能会迫使你 以后花费更高的成本。总的来说，最好的做法是综合考虑最佳实践策略和成本管理以及其它要素， 这在几者之间取得平衡。 使用代码分析和调试工具以便消除瓶颈- 节约成本的最好的做法是：提升系统整体效率，从而 使用更少的硬件和资源。所以，可以使用一些与性能和扩展相关的工具如New Relic以及深度监控工 具，以便在系统扩展时可以减少硬件需求并降低成本。 利用公共云提高灵活性 - 全球的公共云在短短的几年内取得重大进展，这些云不仅非常灵活， 可以使您极大地扩展系统，而且还带有许多额外的有用功能。虽然，使用这些云来处理工作负荷并 不便宜，但是，这些云可以使您在灵活性、功能性以及总体拥有成本（TCO）方面取得平衡。 利用私有云节约更多的成本 - 对于大型系统而言，他们节约成本的策略通常是选择使用私有 云，虽然私有云没有公共云灵活，但是却比公共云要便宜很多。对于大型应用程序而言，由于大型 RAM和CPU的特点限制，若使用公共云的话，会需要花费极其高的成本，所以，使用私有云就变得 十分重要。私有云有足够的容量大小，可以支持多种架构和解决方案，也包括对使用同种硬件的开 发、测试和生产系统提供支持。 如今，互联网系统庞大复杂、功能强大，正如我们所知道的，已经武装了整个世界。要构建好 并管理好这些系统，就必须采用最佳的组合策略，综合使用各种工程、工具、流程和观点，这样才 能够构建出现代化的系统，并且在上述全部五个要素中取得平衡。要想在21世纪取得运维成功和系 统成功的话，关键是要能够理解并合理地关注这五大要素。