

2025 年上半年系统架构设计师考前二十问

一、问：电子政务有哪些类型？对应的具体的应用有哪些？

答：

类型	应用
G2G	基础信息的采集、处理和利用，如：人口信息、地理信息。及政府间：计划管理、财务管理、通信系统、各级政府决策支持。
G2B	政府给企业单位颁发【各种营业执照、许可证、合格证、质量认证】
B2G	企业向政府缴税 企业向政府提供各种商品和服务【含竞/投标】 企业向政府提建议，申诉
G2C	社区公安和水、火、天灾等与公共安全有关的信息 户口、各种证件和牌照的管理
C2G	个人应向政府缴纳的各种税款和费用 个人向政府反馈民意【征求群众意见】 报警服务（盗贼、医疗、急救、火警等）
G2E	政府内部管理系统

二、问：面向信息集成、面向过程集成、面向服务集成与界面集成、数据集成、应用集成、过程集成有什么区别和联系？

答：按照集成的层次进行分类，可将企业内的集成分为：界面集成、平台集成、数据集成、应用集成、过程集成等。

按照集成技术来分为：面向信息的集成、面向过程的集成和面向服务的集成。

对于二者的联系：一般而言，界面集成、平台集成、数据集成、应用集成一般可通过面向信息的集成技术实现，过程集成一般可通过面向过程的集成技术可实现，但其中又常包含接口集成等面向信息的集成技术。当然，应用集成和过程集成也可以通过面向服务的集成技术来实现。

三、问：原型有哪些种类？怎么划分的？

答：在系统开发中，原型可以划分为不同的种类。从原型是否实现功能来分，可以分为水平原型和垂直原型；从原型最终结果来分，可以分为抛弃式原型和演化式原型。

水平原型又称行为原型模型，在一定程度上实现用户交互层的界面布局和界面流转逻辑。水平原型反映了系统的宽度，而没有深入到太多的细节中。它们有助于理解整个系统的能力范围以及特性将如何结合在一起。它们对于展示想法、促进需求讨论以及获得需求和设计决策的认可非常有用。在分析的早期阶段，最常用的是水平原型。

垂直原型又称结构原型，垂直原型用于分析和设计的后期阶段，以深入研究和详细说明特性或功能。

垂直原型在本质上更具技术性，通过真实数据连接到数据库，与现有子系统接口，并精确反映关键功能。

抛弃式原型主要用于界面设计。抛弃式原型基本思路就是开始就做一个简单的界面设计，用来让用户有直观感受，从而可以提出需求，等需求获取到之后，可以把这个界面原型抛弃不用。

而演化式原型，会把原型保留，通过不断的演化，逐步形成最终产品。

四、问：传统五大类风格有哪些？是怎么分类的？

答：

五大架构风格	子风格
数据流风格	批处理、管道-过滤器
调用/返回风格	主程序/子程序、面向对象、层次结构
独立构件风格	进程通信、事件驱动系统（隐式调用）
虚拟机风格	解释器、规则系统
仓库风格	数据库系统、黑板系统、超文本系统

五、问：反规范化有哪些技术手段？反规范化的优缺点是什么？

答：（1）反规范化手段

技术手段	说明
增加派生性冗余列	已有单价和数量列，增加“总价”列
增加冗余列	已有学号列，增加“姓名”列
重新组表	把拆分的表重新组表
分割表	把用户表做水平分割，长沙的用户存在长沙，上海的用户存在上海

（2）反规范化的优缺点

优点：连接操作少，检索快，统计快，需要查的表减少，检索容易。

缺点：

反规范化的缺点	解决方案
数据冗余，需要更大存储空间	无解
插入、更新、删除操作开销更大	无解
数据不一致 可能产生添加、修改、删除异常	1、触发器数据同步 2、应用程序数据同步 3、批处理
更新和插入代码更难写	无解

六、问：微服务的优势是什么？

答：微服务之所以能盛行，必然是有它独特优势的，微服务的优势如下：

(1) 技术异构性：在微服务架构中，每个服务都是一个相对独立的个体，每个服务都可以选择适合于自身的技术来实现。如，要开发一个社交平台，此时，我们可能使用文档型数据库来存储帖子的内容，使用图数据来存储朋友圈的这些关系等，这样可以把每一块的性能都充分发挥出来。同时，在应用新技术时，微服务架构也提供了更好的试验场。因为对于单块的系统而言，采用一个新的语言、数据库或者框架都会对整个系统产生巨大的影响，这样导致我们想尝试新技术时，望而却步。但微服务不同，我们完全可以只在一个微服务中采用新技术，待技术使用熟练之后，再推广到其它服务。

(2) 弹性：弹性主要讲的是系统中一部分出现故障会引起多大的问题。在单块系统中，一个部分出现问题，可能导致整体系统的问题。而微服务架构中，每个服务可以内置可用性的解决方案与功能降级方案，所以比单块系统强。

(3) 扩展：单块系统中，我们要做扩展，往往是整体进行扩展。而在微服务架构中，可以针对单个服务进行扩展。

(4) 简化部署：在大型单块系统中，即使修改一行代码，也需要重新部署整个应用系统。这种部署的影响很大、风险很高，因此不敢轻易的重新部署。而微服务架构中，每个服务的部署都是独立的，这样就可以更快地对特定部分的代码进行部署。

(5) 与组织结构相匹配：我们都知道，团队越大越难管理，同时团队越大也代表系统规模越大、代码库越大，这样容易引起一系列的问题。且当团队是分布式的时候，问题更严重。微服务架构就能很好的解决这个问题，微服务架构可以将架构与组织结构相匹配，避免出现过大的代码库，从而获得理想的团队大小及生产力。服务的所有权也可以在团队之间迁移，从而避免异地团队的出现。

(6) 可组合性：在微服务架构中，系统会开放很多接口供外部使用。当情况发生改变时，可以使用不同的方式构建应用，而整体化应用程序只能提供一个非常粗粒度的接口供外部使用。

(7) 对可替代性的优化：在单块系统中如果删除系统中的上百行代码，也许不知道会发生什么，引起什么样的问题，因为单块系统中关联性很强。但在微服务架构中，我们可以在需要时轻易的重写服务，或者删除不再使用的服务。

七、问：基于软件系统的生命周期，质量属性可以分为哪两类？这两类分别包含了哪些？ 答：软件系统质量属性是一个系统的可测量或者可测试的属性，用来描述系统满足利益相关者需求的程度。基于软件系统的生命周期，可以将软件系统的质量属性分为开发期质量属性和运行期质量属性 2 个部分。

开发期质量属性主要指在软件开发阶段所关注的质量属性，主要包含 6 个方面：

易理解性：指设计被开发人员理解的难易程度。

可扩展性：指软件因适应新需求或需求变化而增加新功能的能力，也称为灵活性。

可重用性：指重用软件系统或某一部分的难易程度。

可测试性：指对软件测试以证明其满足需求规范的难易程度。

可维护性：指当需要修改缺陷、增加功能、提高质量属性时，识别修改点并实施修改的难易程度。

可移植性：指将软件系统从一个运行环境转移到另一个不同的运行环境的难易程度。

运行期质量属性主要指在软件运行阶段所关注的质量属性，主要包含 7 个方面：

性能：指软件系统及时提供相应服务的能力，如速度、吞吐量和容量等的要求。

安全性：指软件系统同时兼顾向合法用户提供服务，以及阻止非授权使用的能力。

可伸缩性：指当用户数和数据量增加时，软件系统维持高服务质量的能力。例如，通过增加服务器来提高能力。

互操作性：指本软件系统与其他系统交换数据和相互调用服务的难易程度。

可靠性：指软件系统在一定的时间内持续无故障运行的能力。

可用性：指系统在一定时间内正常工作的时间所占的比例。可用性会受到系统错误，恶意攻击，高负载等问题的影响。

鲁棒性：指软件系统在非正常情况（如用户进行了非法操作、相关的软硬件系统发生了故障等）下仍能正常运行的能力，也称健壮性或容错性。

八、问：常考的质量属性以及相关设计策略有哪些？

答：（1）性能

性能（performance）是指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理的事件的个数。例如：1.同时支持 1000 并发；2.响应时间小于 1s；3.显示分辨率达到 4K。

代表参数：响应时间、吞吐量 设计策略：优先级队列、资源调度

（2）可用性

可用性（availability）是系统能够正常运行的时间比例。经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。例如：1.主服务器故障，1 分钟内切换至备用服务器；2.系统故障，1 小时内修复；3.系统支持 7×24 小时工作。

代表参数：故障间隔时间 设计策略：冗余、心跳线

（3）安全性

安全性（security）是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。安全性又可划分为机密性、完整性、不可否认性及可控性等特性。例如：1.可抵御 SQL 注入攻击；2.对计算机的操作都有完整记录；3.用户信息数据库授权必须保证 99.9%可用。

设计策略：追踪审计

（4）可修改性

可修改性（modifiability）是指能够快速地对系统性能价格比进行变更的能力。通常以某些具体的变更为基准，通过考察这些变更的代价衡量可修改性。（可扩展性与之相近）例如：1.更改系统报表模块，必须在 2 人·周内完成；2.对 Web 界面风格进行修改，修改必须在 4 人·月内完成。

主要策略：信息隐藏（二义性：良好的封装能够做到信息隐藏，一般归于可修改性策略；信息隐藏也能够体现在安全性当中）

九、问：云原生架构的设计原则和架构模式分别有哪些？

答：云原生架构设计原则：

- (1) 服务化原则：使用微服务
- (2) 弹性原则：可根据业务变化自动伸缩
- (3) 可观测原则：通过日志、链路跟踪和度量
- (4) 韧性原则：面对异常的抵御能力
- (5) 所有过程自动化原则：自动化交付工具
- (6) 零信任原则：默认不信任网络内部和外部的任何人/设备/系统
- (7) 架构持续演进原则：业务高速迭代情况下的架构与业务平衡

云原生架构模式：

- (1) 服务化架构模式：典型代表【微服务】，服务拆分使维护压力大增。
- (2) Mesh 化架构模式：把中间件框架（RPC、缓存、异步消息）从业务进程中分离，由 Mesh 进程完成。
- (3) Serverless 模式：非常适用于事件驱动的数据计算任务。
- (4) 存储计算分离模式：各类暂态数据（如 session）用云服务保存。
- (5) 分布式事务模式：解决微服务模式中多数据源事务问题。
- (6) 可观测架构：包括 Logging、Tracing、Metrics 三个方面。
- (7) 事件驱动架构：本质上是一种应用/组件间的集成架构模式。

十、问：SAAM 和 ATAM 在评估目标、质量属性和评估活动等方面有什么区别？

答：常见的系统体系架构分析方法有 SAAM 和 ATAM。

SAAM (Scenarios-based Architecture Analysis Method) 是一种非功能质量属性的体系架构分析方法，最初用于比较不同的体系架构，分析架构的可修改性，后来也用于其他的质量属性，如可移植性、可扩充性等。

(1) 特定目标：对描述应用程序属性的文档，验证基本体系结构假设和原则。SAAM 不仅能够评估体系结构对于特定系统需求的适用能力，也能被用来比较不同的体系结构。

(2) 评估活动：SAAM 的过程包括五个步骤，即场景开发、体系结构描述、单个场景评估、场景交互和总体评估。

ATAM (Architecture Tradeoff Analysis Method) 是在 SAAM 的基础上发展起来的，主要针对性能、实用性、安全性和可修改性，在系统开发之前，对这些质量属性进行评价和折中。

(1) 特定目标：在考虑多个相互影响的质量属性的情况下，从原则上提供一种理解软件体系结构的能力的方法，使用该方法确定在多个质量属性之间折中的必要性。

(2) 评估活动：分为四个主要的活动领域，分别是场景和需求收集、体系结构视图和场景实现、属性模型构造和分析、折中。

十一、问：被动攻击和主动攻击分别有哪些？

答：被动攻击：收集信息为主，破坏保密性。

攻击类型	攻击名称	描述
被动攻击	窃听（网络监听）	用各种可能的合法或非法的手段窃取系统中的信息资源和敏感信息。
	业务流分析	通过对系统进行长期监听，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等参数进行研究，从而发现有价值的信息和规律。
	非法登录	有些资料将这种方式归为被动攻击方式。

主动攻击：主动攻击的类别主要有：中断（破坏可用性），篡改（破坏完整性），伪造（破坏真实性）

攻击类型	攻击名称	描述
主动攻击	假冒身份	通过欺骗通信系统（或用户）达到非法用户冒充成为合法用户，或者特权小的用户冒充成为特权大的用户的目的。黑客大多是采用假冒进行攻击。
	抵赖	这是一种来自用户的攻击，比如：否认自己曾经发布过的某条消息、伪造一份对方来信等。
	旁路控制 【旁路攻击】	密码学中是指绕过对加密算法的繁琐分析，利用密码算法的硬件实现的运算中泄漏的信息。如执行时间、功耗、电磁辐射等，结合统计理论快速的破解密码系统。
	重放攻击	所截获的某次合法的通信数据拷贝，出于非法的目的而被重新发送。加时间戳能识别并应对重放攻击。
	拒绝服务（DOS）	对信息或其它资源的合法访问被无条件地阻止。

十二、问：恢复块方法与 N 版本程序设计有什么区别？

答：

	恢复块方法	N 版本程序设计
硬件运行环境	单机	多机
错误检测方法	验证测试程序	表决
恢复策略	后向恢复	前向恢复
实时性	差	好

其中：

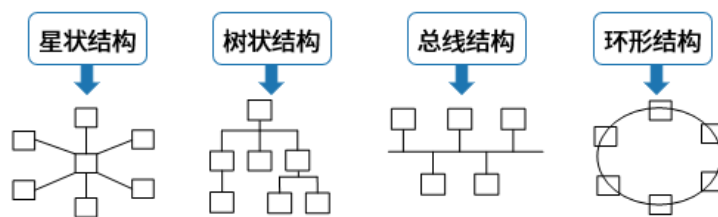
前向恢复：使当前的计算继续下去，把系统恢复成连贯的正确状态，弥补当前状态的不连贯情况。

后向恢复：系统恢复到前一个正确状态，继续执行。

十三、问：常见的局域网拓扑结构有哪些？

答：局域网专用性非常强，具有比较稳定和规范的拓扑结构。常见的局域网拓扑结构有星状结构、树

状结构、总线结构和环形结构。



	特点	缺点
星状结构	网络中的每个结点设备都以中心结点为中心，通过连接线与中心结点相连，如果一个结点设备需要传输数据，它首先必须通过中心结点；中心结点是控制中心，任意两个结点间的通信最多只需两步；传输速度快、网络结构简单、建网容易、便于控制和管理	可靠性低，网络共享能力差，并且一旦中心结点出现故障则导致全网瘫痪。
树状结构	被称为分级的集中式网络；网络成本低，结构简单；在网络中，任意两个结点之间不产生回路，每个链路都支持双向传输，结点扩充方便、灵活，方便寻查链路路径	除叶结点及其相连的链路外，任何一个工作站或链路产生故障都会影响整个网络系统的正常运行
总线结构	网络是将各个结点设备和一根总线相连；网络中所有的结点设备都是通过总线进行信息传输的	作为数据通信必经的总线的负载能力是有限度的，这是由通信媒体本身的物理性能决定的；它的故障将影响总线上每个结点的通信
环状结构	网络中各结点通过一条首尾相连的通信链路连接起来，形成一个闭合环形结构网；网络中各结点设备的地位相同，信息按照固定方向单向流动，两个节点之间仅有一条通路，系统中无信道选择问题	网络不便于扩充，系统响应延时长，且信息传输效率相对较低，任一结点的故障将导致物理瘫痪
网状结构	任何结点彼此之间均存在一条通信链路，任何结点故障不会影响其他结点之间的通信（可靠性高）	网络布线较为繁琐，且建设成本高，控制方法复杂

十四、问：嵌入式数据库有哪些分类？

答：按照数据库存储位置的不同而进行分类是目前广泛采用的分类方法，它可以划分为三类：

	定义	特点
基于内存的数据库 MMDB	基于内存的数据库系统是实时系统和数据库系统的有机结合。	内存数据库是支持实时事务的最佳技术，其本质特征是以其“主拷贝”或“工作版本”常驻内存，即活动事务只与实时内存数据库的内存拷贝打交道。
基于文件的数据库 FDB	基于文件的数据库系统就是以文件方式存储数据库数据，即数据按照一定	这种数据库的访问方式是被动式的，只要了解其文件格式，任何程序都可以直接读取，因此它的

	格式储存在磁盘中。使用时由应用程序通过相应的驱动程序甚至直接对数据文件进行读写。	安全性很低。虽然文件数据库存在诸多弊端，但是，可以满足嵌入式系统在空间、时间等方面的特殊要求。
基于网络的数据库 NDB	基于网络的数据库系统是基于手机 4G/5G 的移动通信基础之上的数据库系统，在逻辑上可以把嵌入式设备看作远程服务器的一个客户端。	嵌入式网络数据库系统的特点是： 无需解析 SQL 语句；支持更多的 SQL 操作；客户端小、无需支持可剪裁性；有利于代码重用。

其中嵌入式网络数据库主要由三部分组成：客户端、通信协议和远程服务器。客户端主要负责提供接口给嵌入式程序，通信协议负责规范客户端与远程服务器之间的通信，还需要解决多客户端的并发问题，远程服务器负责维护服务器上的数据库数据。

十五、问：鸿蒙操作系统整体架构大致是什么样的？

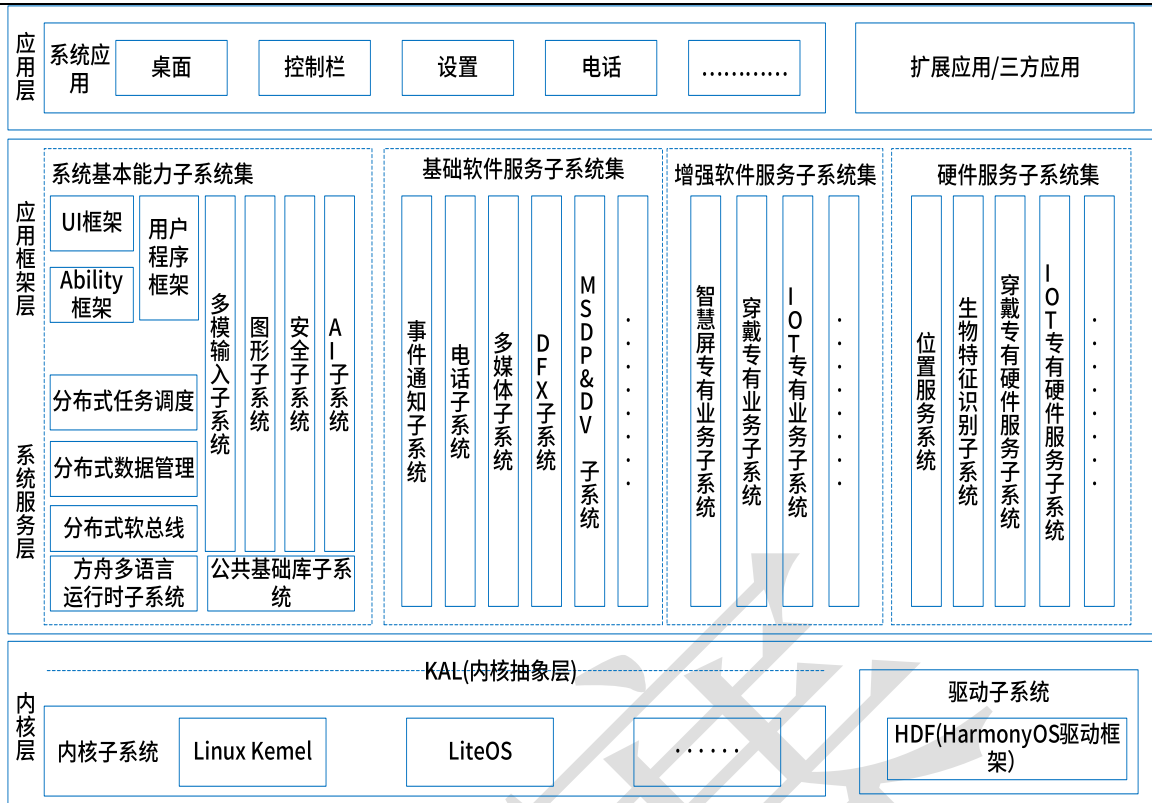
答：Harmony OS 系统架构整体上遵从分层设计，从下向上分为内核层、系统服务层、框架层和应用层。Harmony OS 系统功能按照“系统→子系统→功能/模块”逐步逐级展开，在多设备部署场景下，支持根据实际需求裁剪或增加子系统或功能/模块。

(1) 内核层：鸿蒙系统分为内核子系统和驱动子系统。在内核子系统中鸿蒙系统采用多内核设计，支持针对不同资源受限设备选用合适的 OS 内核；鸿蒙系统驱动框架是鸿蒙系统硬件生态开放的基础，它提供统一外设访问能力和驱动开发、管理框架。

(2) 系统服务层：系统服务层是鸿蒙系统的核心能力集合，通过框架层对应用程序提供服务。包含了系统基本能力子系统集、基础软件服务子系统集、增强软件服务子系统集、硬件服务子系统四个部分。

框架层：框架层为鸿蒙系统应用程序提供 Java/C/C++/JS 等多语言用户程序框架和 Ability 框架，及各种软硬件服务对外开放的多语言框架 API，也为搭载鸿蒙系统的电子设备提供 C/C++/JS 等多语言框架 API。

应用层：应用层包括系统应用和第三方非系统应用，鸿蒙系统应用由一个或多个 FA 或 PA 组成。



此外，Harmony OS 架构的系统安全性主要体现在搭载 Harmony OS 的分布式终端上，可以保证“正确的人，通过正确的设备，正确地使用数据”。这里通过“分布式多端协同身份认证”来保证“正确的人”，通过“在分布式终端上构筑可信运行环境”来保证“正确的设备”，通过“分布式数据在跨终端流动的过程中，对数据进行分类分级管理”来保证“正确地使用数据”。

十六、问：冷备份和热备份的优缺点分别是什么？

答：

备份方式 \ 优缺点	优点	缺点
冷备份	非常快速的备份方法（只需复制文件）；容易归档（简单复制即可）；容易恢复到某个时间点上（只需将文件再复制回去）；能与归档方法相结合，做数据库“最佳状态”的恢复；低度维护，高度安全	单独使用时，只能提供到某一时间点上的恢复；在实施备份的全过程中，数据库必须要作备份而不能做其他工作；若磁盘空间有限，只能复制到磁带等其他外部存储设备上，速度会很慢；不能按表或按用户恢复
热备份	可在表空间或数据库文件级备份，备份的时间短；备份时数据库仍可使用；可达到秒级恢复（恢复到某一时间点上）；可对几乎所有数据库实体做恢复；恢复是快速的	不能出错，否则后果严重；若热备份不成功，所得结果不可用于时间点的恢复；因难于维护，所以要特别小心，不允许“以失败告终”

(1) 冷备份也称为静态备份，是将数据库正常关闭，在停止状态下，将数据库的文件全部备份（复制）下来。

(2) 热备份也称为动态备份，是利用备份软件，在数据库正常运行的状态下，将数据库中的数据文件备份出来。

(3) 完全备份：备份所有数据。

(4) 增量备份：仅备份上一次完全备份之后变化的数据。

(5) 增量备份：备份上一次备份之后变化的数据。

(6) 日志文件：事务日志是针对数据库改变所做的记录，它可以记录针对数据库的任何操作，并将记录结果保存在独立的文件中。

十七、问：分区和分表有什么区别和联系？分区有什么策略？

答：1、分区与分表的区别与联系：

两者都针对数据表，将数据做到分布式，提高数据检索的效率，降低数据库的频繁 I/O 压力值。

分表是真正的生成数据表，是将一张大数据量的表分成多个小表实现数据均衡。

分区并不是生成新的数据表，而是将表的数据均衡分摊到不同的硬盘，系统或是不同服务器存储介质中，实际上还是一张表。

2、分区的优点：

相对于单个文件系统或是硬盘，分区可以存储更多的数据。

数据管理比较方便，比如要清理或废弃某年的数据，就可以直接删除该日期的分区数据即可。

精准定位分区查询数据，不需要全表扫描查询，大大提高数据检索效率。

可跨多个分区磁盘查询，来提高查询的吞吐量。

在涉及聚合函数查询时，可以很容易进行数据的合并。

3、分区的策略：

(1) 范围分区 (RANGE)：就是根据数据库表中某一字段的值的范围来划分分区。如：年份小于 2016 的分成一个区，其它分成另一个区。

(2) 散列分区 (HASH)：散列分区是根据字段的 hash 值进行均匀分布，尽可能的实现各分区所散列的数据相等。

(3) 列表分区 (LIST)：列表分区明确指定了根据某字段的某个具体值进行分区，而不是像范围分区那样根据字段的值范围来划分的。如：长沙、武汉分成一个区，北京一个区。

十八、问：Kappa 架构与 Lambda 架构有什么区别？

答：Kappa 架构与 Lambda 架构的对比

对比内容	Lambda 架构	Kappa 架构
复杂度与开发、维护成本	需要维护两套系统（引擎），复杂度高，开发、维护成本高	只需要维护一套系统（引擎），复杂度低，开发、维护成本低
计算开销	需要一直运行批处理和实时计算，计算开销大	必要时进行全量计算，计算开销相对较小
实时性	满足实时性	满足实时性
历史数据处理能力	批式全量处理，吞吐量大，历史数据处理能力强	流式全量处理，吞吐量相对较低，历史数据处理能力相对较弱
使用场景	直接支持批处理,更适合对历史数据分析查询的场景,期望尽快得到分析结果,批处理可以更直接高效地满足这些需求	不是 Lambda 的替代架构，而是简化，Kappa 放弃了对批处理的支持，更擅长业务本身为增量数据写入场景的分析需求
选择依据	根据两种架构对比分析，将业务需求、技术要求、系统复杂度、开发维护成本和历史数据处理能力作为选择考虑因素。 计算开销虽然存在一定差别，但是相差不是很大，所以不作为考虑因素	

十九、问：数字孪生体是什么？数字孪生体技术有哪些？

答：数字孪生体是现有或将有的物理实体对象的数字模型，通过实测、仿真和数据分析来实时感知、诊断、预测物理实体对象的状态，通过优化和指令来调控物理实体对象的行为，通过相关数字模型间的相互学习来进化自身，同时改进利益相关方在物理实体对象生命周期内的决策。

建模、仿真和基于数据融合的数字线程是数字孪生体的三项核心技术。能够做到统领建模仿真和数字线程的系统工程和 MBSE，则成为数字孪生体的顶层框架技术，物联网是数字孪生体的底层伴生技术，而云计算、机器学习、大数据、区块链则成为数字孪生体的外围使能技术。

1、建模

建模的目的是将我们对物理世界的理解进行简化和模型化。而数字孪生体的目的或本质是通过数字化和模型化，用信息换能量，以使少的能量消除各种物理实体、特别是复杂系统的不确定性。所以建立物理实体的数字化模型或信息建模技术是创建数字孪生体、实现数字孪生的源头和核心技术，也是“数化”阶段的核心。

2、仿真

从技术角度看，建模和仿真是一对伴生体。如果说建模是模型化我们对物理世界或问题的理解，那么仿真就是验证和确认这种理解的正确性和有效性。所以，数字化模型的仿真技术是创建和运行数字孪生体、保证数字孪生体与对应物理实体实现有效闭环的核心技术。

仿真是将包含了确定性规律和完整机理的模型转化成软件的方式来模拟物理世界的一种技术。只要模型正确，并拥有了完整的输入信息和环境数据，就可以基本准确地反映物理世界的特性和参数。

3、其他技术

除了核心的建模仿真技术，目前 VR、AR 以及 MR 等增强现实技术、数字线程、系统工程和 MBSE、物联网、云计算、雾计算、边缘计算、大数据技术、机器学习和区块链技术，仍为数字孪生体构建过程中的内外围核心技术。

二十、问：分布式存储系统架构设计中所使用的分布式存储技术有哪些？

答：分布式存储系统架构设计中所使用的分布式存储技术主要包括四类：

1、集群存储技术。集群存储系统是指架构在一个可扩充服务器集群中的文件系统，用户不需要考虑文件存储在集群中什么位置，仅仅需要使用统一的界面就可以访问文件资源。当负载增加时，只需在服务器集群中增加新的服务器就可以提高文件系统的性能。集群存储系统能够保留传统的文件存储系统的语义，增加了集群存储系统必须的机制，可以向用户提供高可靠性、高性能、可扩充的文件存储服务。

2、分布式文件系统。分布式文件系统是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连。分布式文件系统的设计基于客户机/服务器模式。一个典型的网络可能包括多个供多用户访问的服务器。另外，对等特性允许一些系统扮演客户机和服务器的双重角色。分布式文件系统以透明方式链接文件服务器和共享文件夹，然后将其映射到单个层次结构，以便可以从一个位置对其进行访问，而实际上数据却分布在不同的位置。用户不必再转至网络上的多个位置以查找所需的信息。

3、网络存储技术。网络存储系统就是将“存储”和“网络”结合起来，通过网络连接各存储设备，实现存储设备之间、存储设备和服务器之间的数据在网络上的高性能传输。为了充分利用资源，减少投资，存储作为构成计算机系统的主要架构之一，就不再仅仅担负附加设备的角色，逐步成为独立的系统。利用网络将此独立的系统和传统的用户设备连接，使其以高速、稳定的数据存储单元存在。用户可以方便地使用浏览器等客户端进行访问和管理。

4、P2P 网络存储技术。P2P 网络存储技术的应用使得内容不是存在几个主要的服务器上，而是存在所有用户的个人电脑上。这就为网络存储提供了可能性，可以将网络中的剩余存储空间利用起来，实现网络存储。人们对存储容量的需求是无止境的，提高存储能力的方法有更换能力更强的存储器，另外就是把多个存储器用某种方式连接在一起，实现网络并行存储。相对于现有的网络存储系统而言，应用 P2P 技术将会有更大的优势。P2P 技术的主体就是网络中的 Peer，也就是各个客户机，数量是很大的，这些客户机的空闲存储空间是很多的，把这些空间利用起来实现网络存储。