

## 2020 年 11 月系统架构师真题论文及答案解析

### 试题一 论数据分片技术及其应用

数据分片就是按照一定的规则，将数据集划分成相互独立正交的数据子集。然后将数据子集分布到不同的节点上，通过设计合理的数据分片规则，可将系统中的数据分布在不同的物理数据库中，达到提升应用系统数据处理速度的目的。

请围绕“论数据分片技术及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发软件的项目以及承担的工作

2.Hash 分片、一致性 Hash 分片和按照数据范围分片是三种常用的数据分片方式

3.具体阐述你参与管理和开发的项目，且采用了哪些分片方式，并且具体说明其实现过程和应用效果。

常用的关系型数据库都存在性能瓶颈，即在数据达到一定的量级之后，数据库的性能会显著下降，数据库的读写操作都会随之受到影响。所以就需要对数据库进行优化处理。比如缓存技术，读写分离技术和数据分片技术都可以起到提高性能，缓解单个物理节点的压力。

#### 1.缓存

工作中比较常用的如 Redis 用来缓解数据库的压力。将热点数据预热到缓冲中，避免大量的访问压力直接给到数据库上面，给数据库减轻负担。

#### 2.读写分离

配置实现主从数据库，将请求分为读/写两种类型，读请求走从库(slave)，写类型请求走主库(master)。比如 MySQL 自身提供的主从数据同步方案。主从库之存在较低(可接受范围)的数据同步延迟。

#### 3.数据分片

如果单表/单库存在数据保存的性能问题，可使用分片将保存的数据分散到多个库表中，其中分为水平分片和垂直分片。

#### 4.水平分片

统一类型的数据，分别放到不同的库/表中。每个分片包含了整体的数据集合的一部分。虽然可减轻单节点的访问压力，又迎来了分布式事务的问题。

#### 5.垂直分片

存在一个宽表(即包含过多字段的表)，其中某几个字段属于热点数据，客户端请求某一条记录，大部分情况下都是要获取这条记录中的某几个热点字段。这个时候，将这张表拆分为主表和从表两张表，热点数据单独成表(从表)，这样数据访问/更新会避免在宽表上的大量操作。提前合计好对应的主从表。按照不同的业务模块拆分数据库，这样可以适当的减少单个服务器的压力。

6.三种数据分片方式：hash 方式，一致性哈希(consistent hash)，按照数据范围(range based)

#### ①哈希分片

概念：按照数据的某一特征来计算哈希值，并将哈希值与系统中的节点建立映射关系。

优点：简单易于实现

缺点：很难解决数据不均衡问题，再增加一个机器，每个机器对应的一个 hash 值的区域就发生改变。

补充：假设这里面是按员工的薪水进行计算 hash 值，实际人群中，可能处于平均薪水 10k 左右的人比较多，高薪水的人比较少这导致某些机器上的数据很大，导致大量的数据集中到一个物理节点上。

## ②一致性哈希

概念：一致性 hash 相当与一个环。所有的数据都在这个环上，每个机器相当于环的一截，相比于上述的 hash 方式，一致性 hash 方式需要维护的元数据额外包含了节点在环上的位置。

优点：简单易于实现，在增删数据的时候只会影响到 hash 环上相邻的节点，不会发生大规模的数据迁移。

缺点：增加节点的时候，只能分摊一个已存在节点的压力

补充：

在实际工程中，一般会引入虚拟节点（virtual node）的概念。即不是将物理节点映射在 hash 环上，而是将虚拟节点映射到 hash 环上。虚拟节点的数目远大于物理节点，因此一个物理节点需要负责多个虚拟节点的真实存储。操作数据的时候，先通过 hash 环找到对应的虚拟节点，再通过虚拟节点与物理节点的映射关系找到对应的物理节点。

## 3.按照数据范围

概念：就是按照关键值划分成不同的区间，每个物理节点负责一个或者多个区间。其实这种方式跟一致性 hash 有点像，可以理解为物理节点在 hash 环上的位置是动态变化的。

优点：

当达到这个阈值之后就会分裂成两个块。这样做的目的在于当有节点加入的时候，可以快速达到均衡的目的

缺点：

在数据可修改的情况下，如果块进行分裂，那么元数据中的区间信息也需要同步修改。

补充：

range based 这种数据分片方式应用非常广泛，比如 MongoDB, PostgreSQL, HDFS

## 4.比较

如果一个节点负责的数据只有一个区间，range based 与没有虚拟节点概念的一致性 hash 很类似；如果一个节点负责多个区间，range based 与有虚拟节点概念的一致性 hash 很类似。

分片方式	映射难度	元数据	节点增删	数据动态均衡
哈希方式	简单	非常简单，几乎不用修改	需要迁移的数据比较多	不支持
一致性哈希	简单	比较简单，取决于节点规模，几乎不用修改	增删节点的时候只影响hash环上相邻节点，但不能使所有节点都参与数据迁移过程	不支持
一致性哈希 (虚拟节点)	中等	稍微复杂一些，主要取决于虚拟节点规模，很少修改	需要迁移的数据比较少，且所有节点都能贡献部分数据	弱支持（修改虚拟节点与物理节点映射关系）
范围分片	较为复杂	取决于每个块的大小，一般来说规模较大，且修改频率较高	需要迁移的数据比较少，且所有节点都能贡献部分数据	支持，且比较容易

5.数据分片需要按照一定的规则，不同的分布式应用有不同的规则，但都遵循同样的原则：按照最主要、最频繁使用的访问方式来分片。

1.具体如何划分原始数据集

2.当原问题的规模变大的时候，能否通过增加节点来动态适应

3.当某个节点故障的时候，能否将该节点上的任务均衡的分摊到其他节点

4.对于可修改的数据（比如数据库数据），如果某节点数据量变大，能否以及如何将部分数据迁移到其他负载较小的节点，及达到动态均衡的效果

5.元数据的管理（即数据与物理节点的对应关系）规模

6.元数据更新的频率以及复杂度

## 试题二 论云原生架构及其应用

论云原生架构及其应用论题，依次从以下三个方面进行论述：

1.概要叙述你参与管理和开发的软件项目以及承担的主要工作。

2.服务化、强性、可观测性和自动化是云原生架构重复的四类设计原则，请简要对这四类设计原则的内涵进行阐述。

3.具体阐述你参与管理和开发的项目是如何向采用云原生架构的，并且围绕上述四类设计原则详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

1.云原生是一种方法，用于构建和运行充分利用云计算模型优势的应用。云计算不再将重点放在资本投资和员工上来运行企业数据中心，而是提供无限制的按需计算能力和根据使用情况付费的功能，从而重新定义了几所有行业的竞争格局。

2.IT 开销减少意味着入行的壁垒更低，这一竞争优势使得各团队可以快速将新想法推向市场，这就是软件正在占据世界，并且初创公司正在使用云原生方法来颠覆传统行业的原因。

3.但是，企业需要一个用于构建和运行云原生应用和服务的平台，来自动执行并集成 DevOps、持续交付、微服务和容器等概念。

4.从技术角度看，云原生分两大部分，一部分是遵循微服务化和容器化原则的云原生应用，另一个部分是用于构建和运行云原生应用的云原生平台。

5.云原生应用和云原生平台，共同构成了一个云原生的完整体系，在这个体系上，可以实践敏捷开发、DevOps、容器编排，微服务和容器化等理论和方法。

DevOps：

开发和运维之间保持流程连续的协作方法，目标是快速、频繁且更可靠地构建、测试和发布软件。

容器化：

与虚拟机相比，容器能同时提供更好的效率和启动速度。每个容器都具有唯一的可写文件系统和资源配额。创建和删除容器的开销较低，在单个虚拟机上能通过容器化充分利用物力资源，这使的容器成为部署微服务的完美工具。

微服务

是将大型应用作为小型服务集合进行开发的架构方法，其中每个服务都可实现业务功能，在自己的流程中运行并通过 HTTP API 进行通信。每个微服务都可以独立于其它服务进行部署、升级、扩展和重新启动，通常作为自动化系统的一部分运行，可以在不影响最终客户的情况下频繁更新正在使用中的应用。

## 试题三 论软件测试中缺陷管理及其应用

围绕“论软件测试中缺陷管理及其应用”论题，依次从以下三个方面进行论述。

- 1.概要叙述你参与管理和开发的软件项目以及承担的工作
- 2.详细论述常见的缺陷种类及级别，论述缺陷管理和基本流程
- 3.结合你具体参与管理和开发的实际项目，说明是如何进行缺陷管理的。请具体说明实施过程及应用效果。

#### 1.软件缺陷的定义

软件缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。

IEEE729-1983 对缺陷的标准定义：从产品内部看，缺陷是软件产品开发或维护过程中存在的错误、毛病等各种问题；

从产品外部看，缺陷是系统所需要实现的某种功能的失效或违背。

导致软件缺陷产生的原因也是多种多样的，软件工程过程中的人、过程、工具都有可能产生软件缺陷，过程中的每一个环节都有可能产生缺陷

#### 2.软件缺陷的分类

功能没有实现或与需求规格说明不一致；

界面、消息、提示、帮助不够准确或误导用户；

屏幕显示、打印结果不正确；

软件无故退出或没有反应；

边界条件未做处理，输入错误数据没有提示和说明；

运行速度慢或占用资源过多；

与常用的交互软件不兼容；

#### 3.缺陷管理的目的是：

对各个阶段测试发现的缺陷进行跟踪管理，以保证各级缺陷的修复率达到标准，主要实现以下目标：

- (1) 保证信息的一致性；
- (2) 保证缺陷得到有效的跟踪，缩短沟通时间，解决问题更高效；
- (3) 收集缺陷数据并进行数据分析，作为缺陷度量的依据。

#### 4.参与缺陷管理的角色

测试工程师：发现和回归 BUG

测试经理：判断 BUG 的有效性

开发经理：分配 BUG

开发工程师：修改 BUG

#### 5.缺陷来源

Requirement：由于需求的问题引起的缺陷(需求不完全或逻辑错误)

Architecture：由于构架的问题引起的缺陷（登录 session 失效）

Design：由于设计的问题引起的缺陷（图片大小，页面元素显示问题等

Code：由于编码的问题引起的缺陷

Test：由于测试的问题引起的缺陷（软件测试的设计与实施发生错误。特别是系统级的功能测试）

Integration：由于集成的问题引起的缺陷

## 6.缺陷严重性和优先级

### 缺陷的严重性 说明

0 级（致命） 最严重等级，缺陷导致系统任何一个主要功能完全丧失、用户数据受到破坏、系统崩溃、悬挂、死机等

1 级（严重） 系统的主要功能部分丧失、数据不能完全保存，系统的次要功能完全丧失，系统所提供的功能或服务收到明显影响

2 级（一般） 系统的次要功能没有完全实现，但不影响用户的正常使用。例如，提示信息不太准确；或用户界面差、操作时间稍长等问题

3 级（微小） 操作者不方便或遇到麻烦，但不影响功能的操作和执行，如字体不美观、按钮大小不很合适、字体排列不对齐等一些小问题

### 缺陷的优先级 说明

立即解决（p1 级） 缺陷导致系统几乎不能完全运行、使用，或严重妨碍测试的执行，需立即修正、尽快修正

高优先级（p2 级） 缺陷严重，影响测试，需要优先考虑修正，如不超过 24 小时修正

正常排队（p3 级） 缺陷需要修正，但可以正常排队等待修正

低优先级（p4 级） 缺陷可以在开发人员有时间的时候被修正，如果没时间可以不修正

## 试题四 论企业集成架构设计及应用

围绕“论企业集成架构设计及应用”为题

1.概要叙述你参与的软件开发项目的及承担的主要工作

2.详细说明三类企业集成架构设计技术分别要解决的问题及其含义，并阐述每种技术具体包含了哪些集成架构。

3.根据你所参与的项目，说明用了哪些企业集成架构设计技术，实施效果如何。

1.集成是企业信息化和谐发展的阳光大道，要走上这条大道则需要建立企业的集成架构。

2.企业集成架构的问题不同于业务系统，即它更为关注企业的 IT 软件基础设施。

目前的“信息孤岛”、“流程割裂”、“企业运行全局不可见”等问题已经愈加明显，直接制约企业的业务发展。

3.打破信息孤岛的方法是，建立企业级的数据总线，连接不同的业务系统，业务系统数据通过统一的翻译处理后形成企业内部甚至是企业间/行业间的“世界语”，直接简化系统之间的连接方式和数据处理的紧密耦合，实现企业的数据同步和共享。

4.企业集成架构是业务架构、数据架构等的技术基础，只有在制定了企业 IT 环境的“和谐制度”和可持续发展策略之后，其他架构才具有复用价值和意义。

5.针对于企业应用，依据不同的关注点，架构可以分为如下几类：

业务架构（Business Architecture）：关注于业务及其流程；

应用架构（Application Architecture）：关注于应用系统设计；

基础架构（Infrastructure Architecture）：关注于基础技术；

数据架构（Data Architecture）：关注于数据存储及其规划；

6.从发展和动态的角度看，集成技术将使企业有手段和机会不断调整和优化企业业务流程的运行，业务系统的专业化程度将增强，企业的业务流程将逐步梳理以趋于最优，基于集成架构的企业业务流程运行将更趋向自动化。业务流程的自动化体现业务系统之间的和谐共处。对于企业而言，业务流程自动化是追求的目标，也是企业信息化发展的方向。

7.企业集成架构发展的基石是开放的标准和规范，从共享和协同的特征看，企业集成架构应采用 SOA 的设计理念，在尽量不改变企业原有 IT 系统的前提下，通过 SOA 技术，封装不同业务系统对外提供的服务。

【软考达人】

# 软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题
- 4、免费督考群



**微信扫一扫，立马获取**



**最新免费题库**



**备考资料+督考群**

PC版题库：[ruankaodaren.com](http://ruankaodaren.com)