

论文复习说明

论文写作概述:

时间: 120分钟; 具体什么时间段以打印准考证为准。

题目: 四选一, **必有一题考察软件架构**, 每个题目下会有概述和子题目, 要认真看。

考察形式: 机试, 约2500字。

题目类型:

- 1、软件架构, 自从2020年以来, 更偏向于去考察具体的架构, 而不是之前的宏观上的架构风格、架构评估等内容。尤其喜欢考察改版后的八大架构, 如云原生、微服务、安全架构等。
- 2、系统开发, 这块也是每年必考, 软件工程全生命周期都有可能考察, 而且也是更具体, 比如具体的开发方法、开发模型, 以及需求分析、设计、测试、运维等全过程。
- 3、系统可靠性、安全性、容错技术等。
- 4、企业应用集成、企业集成平台等。
- 5、其他: 项目管理、数据库等。

历年真题汇总分析

年份 ⁺	试题一 ⁺	试题二 ⁺	试题三 ⁺	试题四 ⁺
202405 ⁺	论大数据 lambda 架构 ⁺	论模型驱动架构设计方法及其应用 ⁺	单元测试 ⁺	论云上自动化运维及其应用 ⁺
2023 ⁺	可靠性分析与评价方法 ⁺	面向对象分析 ⁺	多数据源集成 ⁺	边云协同 ⁺
2022 ⁺	论基于构件的软件开发方法及其应用 ⁺	论软件维护方法及其应用 ⁺	论区块链技术及应用 ⁺	论湖仓一体架构及其应用 ⁺
2021 ⁺	论面向方面的编程技术及其应用 ⁺	论系统安全架构设计及其应用 ⁺	论企业集成平台的理解与应用 ⁺	论微服务架构及其应用 ⁺
2020 ⁺	论企业集成架构设计及其应用 ⁺	论软件测试中缺陷管理及其应用 ⁺	论云原生架构及其应用 ⁺	论数据分片技术及其应用 ⁺
2019 ⁺	论软件设计方法及其应用 ⁺	论软件系统架构评估及其应用 ⁺ I	论数据湖技术及其应用 ⁺	论负载均衡技术在 web 系统中的应用 ⁺
2018 ⁺	论软件开发过程 RUP 及其应用 ⁺	论软件体系结构的演化 ⁺	论面向服务架构设计及其应用 ⁺	论 NoSQL 数据库技术及其应用 ⁺
2017 ⁺	论软件系统建模方法及其应用 ⁺	论软件架构风格 ⁺	论无服务器架构及其应用 ⁺	论软件质量保证及其应用 ⁺
2016 ⁺	论软件系统架构评估 ⁺	论软件设计模式及其应用 ⁺	论数据访问层设计技术及其应用 ⁺	论微服务架构及其应用 ⁺
2015 ⁺	论应用服务器基础软件 ⁺	论软件系统架构风格 ⁺	论面向服务的架构及其应用 ⁺	论企业集成平台的技术与应用 ⁺

常见问题

问题1：论文字数范围，各部分字数如何分配？

答：首先，答题卡最大字数是330+2750，肯定不能超过答题卡；其次，建议摘要300字左右，正文2000-2700字都可以，建议2200字，写到答题卡最后一页即可；最后字数分配建议摘要300字+项目背景500字+主体1200字+结尾500字。

问题2：我没有做过系统架构设计师怎么办？

答：如果你是专业人员，虽然不是系统架构设计师，但跟过项目，那么就从系统架构设计师的角度去写熟悉的项目；如果你是完全零基础，行外人，就参考范文，找一个觉得合适的项目背景改写。

问题3：我做的是公司内部的项目，没有客户怎么办？

答：不要在论文里写这是内部项目，当成普通项目写，如某年某月，公司要做什么项目，任命我为系统架构设计师即可，最后收尾仍然写客户一致满意，按照一般项目来写。

问题4：是否需要写论文题目？

答：答题卡上不能写论文题目，直接在答题卡那里将你选择的题号涂黑即可。

常见问题

问题5：数字、英文字母、标点符号该怎么写？

答：遵循小学生作文写作原则，两个数字占一个格子；字母按自然大小连起来写，看着不拥挤，使其自然占几个格子；标点符号要占一个格子，并且每行开头第一个格子不能有标点符号。以上这些都算字数，因为论文答题纸是标识格子的数量，没人会去数你到底写了多少汉字，有多少是空格。

问题6：项目背景如何构造才比较合理？

- ◆关于项目时间：建议离考试时间五年以内，持续时间8个月以上，并且已完成的项目；如果说项目真的是比较久远的，或者持续时间较短的，但是规模是足够的，请自己灵活修改时间。
- ◆关于项目级别：建议是省市级、集团级，不要是县级、社区级等，如果你做的项目规模是符合要求的，真的是县级的真实项目，那没有问题。
- 是否要写真实的项目名称：不需要，也不能写，只能写某地级市，某医院，某公司。
- ◆关于项目团队成员：可省略，为了保证真实性，也可以简单带过，例如有需求分析人员3人，开发人员8人，测试人员5人等等。
- ◆关于项目金额：要和项目时间还有项目团队成员对应，如果项目组一共有20人，按每人每月2万算，一个月需要40万，再乘以持续时间假设是15个月，那么项目金额在600万左右是合理的，但不要写整数，可以写565万，618万等字眼。
- ◆特别注意：如果是自己编的项目，请不要构造太大的项目，项目金额最好在500万左右。

搭建自己的万能模板

1. 合适的项目

选择中大型商业项目，一般金额在200W以上，研发周期在8个月以上的项目。

推荐项目

政府或大型信息系统项目：各国企、事业单位、军方、医院、银行、股份公司大企业的ERP、OA等；云计算、大数据等各种软件及信息系统。

不能选项目

小型企业项目：如进销存系统、图书管理系统、单机版系统。

尚未完成的项目：一定要已经完成并上线运行，并最好在三年内。

纯建网站项目：如建设企业或政府门户网站，只有静态网页链接介绍，没有后台大型应用的。

硬件项目：如综合布线、安防、视频会议等。

纯技术项目：如数据升级迁移、内部技术研究等。

建议：在推荐的范围内有限选择自己做的或熟悉的中大型项目，如果是零基础没有做过项目的学员，就参考后面的范文，用别人的项目来模仿。

搭建自己的万能模板

2. 提前准备论文摘要

摘要自己参考范文或者下面的格式，根据自己选择的项目准备一个就可以了。建议逻辑上分两段（但是实际写作不要写两段，因为格子可能不够），第一段是通用的介绍项目背景，第二段是根据不同的论文题目发挥的，简单回应子题目并介绍论文结构。

注意：摘要最多只有330个格子，千万不能超，建议写300个字即可。

包含内容：项目名称、项目金额、项目历时、项目简介、我的责任；本文讨论主题概括。

摘要模板（时间+项目+项目简介+投入+历时+成功交付客户好评+结合具体题目说明本文结构）：

2021年3月，我参与了某航天研究所某型号卫星的全数字仿真验证平台项目的建设，并担任系统架构设计师，负责系统架构设计工作。该系统包括虚拟目标机仿真、动力学模型仿真、同步时序控制三大功能模块，能够模拟卫星在太空中运行所需的所有硬件及外部力学环境，从而可以在虚拟平台中充分测试卫星软件的功能及性能以提高卫星软件的可靠性。该项目总投入565万元人民币，历时15个月，于2022年06月正式交付运行至今，受到了客户的一致好评。本文结合笔者的实际工作经验就该项目的……（根据不同论文题目去简要概括本文内容）。

搭建自己的万能模板

其他论文摘要格式：

(1) 本文讨论.....系统项目的.....(指的是项目主题, 例如进度管理等), 该系统是由某单位建设的, 投资多少万, 系统是用来做什么的(项目背景, 简单功能等)。在本文中, 首先讨论了.....

(过程、方法、措施), 最后.....(主要是不足之处/如何改进/特色之处/发展趋势等)。在本项目的开发过程中, 我主要担任了.....(在本项目中的角色)。

(2) 根据.....需求(项目背景), 我所在的.....组织了.....项目的开发。该项目.....(项目背景、简单功能介绍)。在该项目中, 我担任了.....(角色)。我通过采取.....(过程、方法、措施等), 使项目圆满成功, 得到了用户的一致好评。但通过项目的建设, 我发现.....(主要是不足之处/如何改进/特色之处/发展趋势等)

(3) ...年...月, 我参加了.....项目的开发, 担任.....(角色)。该项目投资多少, 建设工期是多少, 该项目是为了(项目背景、功能介绍)。本文结合作者的实践, 以.....项目为例, 讨论.....(论文主题), 包括.....(过程、方法、措施)。

(4)是.....(戴帽子, 讲述论文主题的重要性, 比如进度的重要性)。本文结合作者的实践, 以.....项目为例, 讨论.....(论文主题), 包括.....(过程、方法、措施)。在本项目的开发过程中, 我担任了.....(角色)。

搭建自己的万能模板

3.提前准备项目背景

项目背景及过渡部分建议500-600字左右, 不能超过太多, 需要提前准备好, 尽量通用化, 不要和论文主题相关, 这样考试时候无论什么论文主题都可以直接默写, 只需要在最后写一段过渡语句, 过渡到下一个论点。

内容包括: 项目开发的原因、你的岗位职责、项目开发周期及规模、项目功能组成介绍、项目技术(可省略)。

项目背景模板(为什么要做这个项目+项目功能和技术介绍+回应子题目并过渡到主体):

在航天卫星的研制过程中, 一颗航天卫星只有一套配套的硬件设备, 无法满足一个开发团队的测试需求, 同时因为航天卫星硬件设备造价十分昂贵, 测试人员也无法进行一些非常规的极限测试, 以防止损坏硬件设备, 以上种种, 将会造成对航天卫星软件的测试不充分和不彻底, 有可能导致卫星研制失败。为了防止这种情况的发生, 针对某重要型号的航天卫星的研制, 该航天研究所领导决定使用技改经费投资建设一套全数字仿真验证平台, 以纯软件的方式模拟卫星在太空中运行所需的全部硬件及外动力学环境, 在虚拟平台中运行卫星软件和动力学模型并进行详细彻底的测试。

搭建自己的万能模板

我所在的公司成功中标该项目，并于 2021 年 03 正式启动该项目的建设工作，我被任命为该项目的系统架构设计师，负责系统架构设计工作。该项目总投入 565 万元人民币，建设周期从 2021 年 3 月 10 日至 2022 年 6 月 30 日止，历时 15 个月。系统采用两台联想 Think Station P510 搭载软件的运行，考虑到对于性能及执行效率方面的要求，使用 C 语言模拟 CPU 指令集及外部设备驱动，使用 [labview](#) 图形化语言搭建地面遥测遥控终端界面。项目的主要建设内容包括三大模块：一、虚拟目标机仿真，完成对卫星软件运行所需的全部硬件环境的模拟，包括内核模拟、片内外设模拟、板级外设模拟等子系统；二、动力学模型仿真，完成对动力学模型运行所需的环境的模拟，包括动力学模型运行环境模拟、故障注入模拟、动力学同步数据控制与显示等子系统；三、同步时序控制，控制卫星软件及动力学模型的启停以及二者之间的时序同步，包括总控台监控、同步时序控制、超实时运行控制等子系统。←

笔者所在的公司虽然在其他型号航天卫星的全数字化仿真验证平台项目上取得成功，但由于卫星型号及用途不同，所涉及的硬件及使用标准也不同，对应的动力学模型也完全不同，而且由于严格的保密性要求，团队成员无法获取全部的卫星型号资料，这无疑加大了项目开发的难度和风险。于是笔者决定在……（过渡段，可以自行参考论文来写，需要回应子题目并引出正文）。↵

搭建自己的万能模板

4.正文写作

正文应该按照论文题目和子题目的要求来写作，并且一定要回应论文子题目。

正文需要写1200字左右。正文部分不在万能模板里，需要根据不同的题目来准备，详细见第三章内容，本章可以先略过，先准备好自己的万能模板。

搭建自己的万能模板

5.提前准备结尾

结尾是个非常有意思的部分，从其本身意义来说，是让你总结项目收获和不足的，然后另一方面，还是你整体补救论文的最后一步。如果你到最后发现字数不够，结尾就需要多写一些，如果你发现字数多了，结尾就要少写一点，如果你觉得前面写的很差，结尾就一定要重视。

结尾可以写400-600字左右，是对整体论文的总概。

包括内容：项目上线及运行效果、客户评价、项目收获、项目不足和解决思路。

经过近 15 个月的项目开发，该型号航天卫星的全数字仿真验证系统顺利投入使用，协助客户对卫星软件进行全面的功能和性能上的测试，运行至今客户反馈良好。该系统由于保密性高，性能要求高，技术实现难度高，项目建设周期长等原因，建设过程困难重重。但由于笔者及项目团队成员十分重视项目的……（回应具体论文题目），最终保证了该项目按质按量顺利交付。

当然，在本项目中，还有一些不足之处，比如：……（自己去想一些小问题，切忌，别出现什么大问题），不过，经过我后期的纠偏，并没有对项目产生什么影响。在后续的学习和工作中，我将不断的充电学习，和同行进行交流，提升自己的专业技术水平，更好的完成系统架构设计的工作。

2024年11月论文1

在2024年05月，文老师成功预测中了大数据架构，精准命中lambda架构，这也说明，架构论文的侧重点确实会倾向于改版后的八大架构了。

让我们来看看第二版下篇内容，分别是：信息系统架构、层次式架构、云原生架构、面向服务架构、嵌入式系统架构、通信系统架构、安全架构、大数据架构。

一共八章，结合架构本身科目特性以及历年真题考察情况，文老师预测2024年11月论文考试，很有可能考到的具体架构是：层次式架构、云原生架构。

云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化的剥离，从而让云设施接管应用中原有的大量非功能特性(如弹性、韧性、安全、可观测性、灰度等),使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。

云原生的代码通常包括三部分：业务代码、三方软件、处理非功能特性的代码。从业务代码中剥离大量非功能性特性(不会是所有，比如易用性还不能剥离)到IaaS和PaaS中，从而减少业务代码开发人员的关注范围，通过云厂商的专业性提升应用的非功能性能力。

具备云原生架构的应用可以最大程度利用云服务和提升软件交付能力，进一步加快软件开发。其特点包括：代码结构发生巨大变化、非功能性特性大量委托、高度自动化的软件交付。

2024年11月论文1

云原生架构原则

服务化原则：拆分为微服务架构、小服务架构，分别迭代。

弹性原则：系统的部署规模可以随着业务量的变化而自动伸缩，无须根据事先的容量规划准备固定的硬件和软件资源。

可观测原则：通过日志、链路跟踪和度量等手段，使得一次点击背后的多次服务调用的耗时、返回值和参数都清晰可见。

韧性原则：当软件所依赖的软硬件组件出现各种异常时，软件表现出来的抵御能力。

所有过程自动化原则：一方面标准化企业内部的软件交付过程，另一方面在标准化的基础上进行自动化，通过配置数据自描述和面向终态的交付过程，让自动化工具理解交付目标和环境差异，实现整个软件交付和运维的自动化。

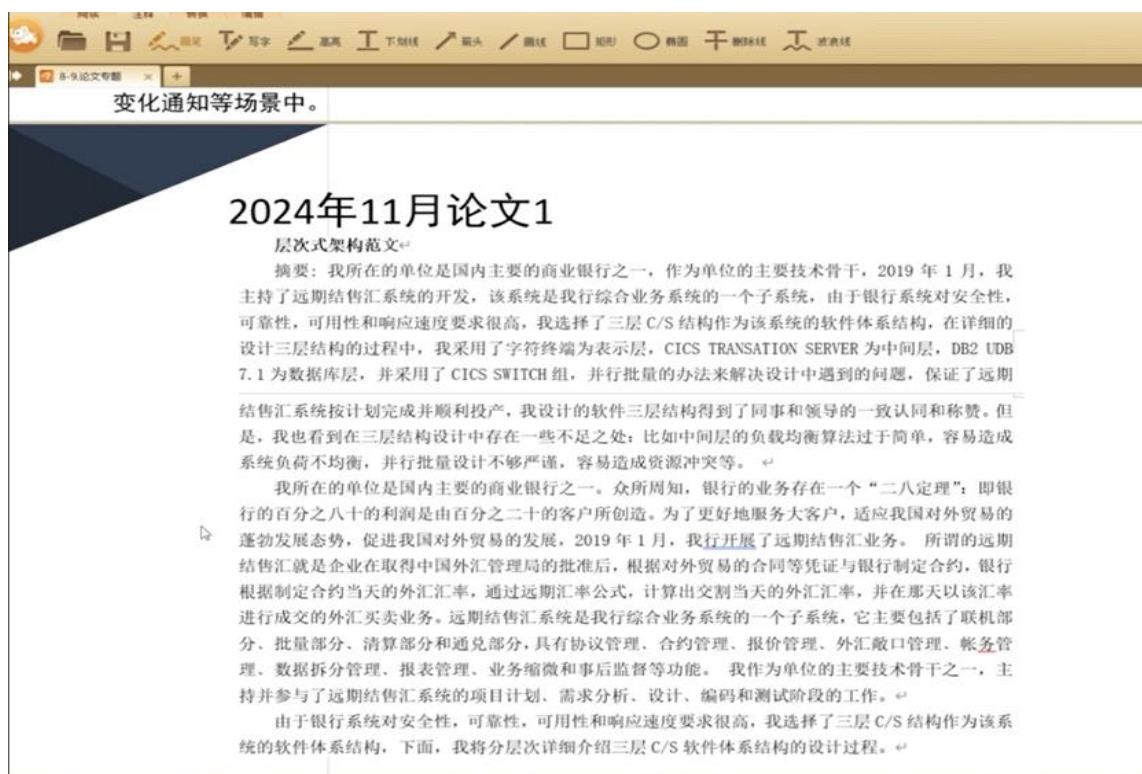
零信任原则：默认情况下不应该信任网络内部和外部的任何人/设备/系统，需要基于认证和授权重构访问控制的信任基础，以身份为中心。

架构持续演进原则：云原生架构本身也必须是一个具备持续演进能力的架构。

2024年11月论文1

主要架构模式

1. 服务化架构模式：典型模式是微服务和小服务模式。通过服务化架构，把代码模块关系和部署关系进行分离，每个接口可以部署不同数量的实例，单独扩缩容，从而使得整体的部署更经济。
2. Mesh化架构模式：把中间件框架(如RPC、缓存、异步消息等)从业务进程中分离，让中间件SDK与业务代码进一步解耦，从而使得中间件升级对业务进程没有影响，甚至迁移到另外一个平台的中间件也对业务透明。分离后在业务进程中只保留很“薄”的Client部分，Client通常很少变化，只负责与Mesh进程通信，原来需要在SDK中处理的流量控制、安全等逻辑由Mesh进程完成。
3. Serverless模式：将“部署”这个动作从运维中“收走”，使开发者不用关心应用运行地点、操作系统、网络配置、CPU性能等，从架构抽象上看，当业务流量到来/业务事件发生时，云会启动或调度一个已启动的业务进程进行处理，处理完成后云会自动关闭/调度业务进程，等待下一次触发，也就是把应用的整个运行都委托给云。
4. 存储计算分离模式：。在云环境中，推荐把各类暂态数据(如session)、结构化和非结构化持久数据都采用云服务来保存，从而实现存储计算分离。
5. 分布式事务模式：大颗粒度的业务需要访问多个微服务，必然带来分布式事务问题，否则数据就会出现不一致。架构师需要根据不同的场景选择合适的分布式事务模式。
6. 可观测架构：可观测架构包括Logging、Tracing、Metrics 三个方面，其中Logging提供多个级别的详细信息跟踪，由应用开发者主动提供；Tracing提供一个请求从前端到后端的完整调用链路跟踪，对于分布式场景尤其有用；Metrics则提供对系统量化的多维度度量。
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。可用于服务解耦、增强服务韧性、数据变化通知等场景中。



2024年11月论文1

1、CICS SWITCH 组。众所周知，银行系统对于安全性，可靠性，可用性和响应速度要求很高，特别是我行最近进行了数据集中，全国只设两个数据中心，分别在 XX 和 YY 两个地方，这样对以上的要求就更高了，为了保障我行的安全生产，我采用了 CTS SWITCH 组技术，所谓的 CICS SWITCH 组，

就是一组相同的 CTS，每个 CTS 上都有相同的业务逻辑，共同作为中间层，消除了单点故障，确保了系统的高度可用性。为了简化系统的设计和缩短通讯时间，我采用了简单的负载均衡算法，比如这次分配给第 N 个 CTS，下次则分配给第 N+1 个 CTS，当到了最后一个，就从第一个开始；为了更好地实现容错，我采用了当第 N 个 CTS 失效的时候，把它正在处理的业务转到第 N+1 个上面继续处理，这样大大增加了系统的可用性，可以为客户提供更好的服务；此外，我还采用了数据库连接池的技术，大大缩短了数据库处理速度，提高了系统运行速度。

2、并行批量。银行系统每天都要处理大量的数据，为了确保白天的业务能顺利进行，有一部分的业务处理，比如一部分内部户帐务处理，或者代理收费业务和总帐与分户帐核对等功能就要到晚上批量地去处理，但是，这部分数据在数据集中之后就显得更加庞大，我行以前采用串行提交批量作业的办法，远远不能适应数据中心亿万级的数据处理要求，在与其他技术骨干讨论之后，并经过充分的论证和试验，我决定采用了并行批量的技术，所谓的并行批量，就是在利用 IBM 的 OPC (Tivoli Operations, Planning and Control) 技术，把批量作业按时间和业务处理先后顺序由操作员统一提交的基础上，再利用 DB2 的 PARTITION 技术，把几个地区分到一个 PARTITION 里面分别处理，大大提高了银行系统的数据处理速度，确保了远期结售汇系统三层结构的先进性。在并行批量的设计过程中，我考虑到批量作业有可能因为网络错误或者资源冲突等原因而中断，这样在编写批量程序和作业的时候必须支持断点重提，以确保生产的顺利进行。

2024年11月论文1

由于我软件三层结构设计得当，并采取了有效的措施去解决设计中遇到的问题，远期结售汇系统最后按照计划完成并顺利投产，不但保证了系统的开发性开放性、可用性和互用性，取得了良好的社会效益和经济效益，而且我的软件三层结构设计得到了同事和领导的一致认同与称赞，为我行以后系统的开发打下了良好的基础。

在总结经验的同时，我也看到了我在软件三层结构设计中的不足之处：首先，负载算法过于简单，容易造成系统的负荷不均衡；由于每个业务的处理时间不一样，有的可能差距很远，简单的顺序加一负载分配算法就容易造成负载不均衡，但是如果专门设置一个分配器，则增加了一次网络通讯，使得系统的速度变慢，这样对响应速度要求很高的银行系统来说也是不可行的，于是我决定采用基于统计的分配算法，即在收到请求的时候，根据预先设定的权值，按概率，直接分配给 CTS。其次，由于批量作业顺序设计得不够严谨等各种原因，容易造成资源冲突；在远期结售汇系统运行了一段时间之后，数据中心的维护人员发现了，系统有的时候会出现资源冲突现象，在经过仔细的分析之后，我发现，由于每天各个业务的业务量大小不一样，顺序的两个作业之间访问同一个表的时候便会产生资源冲突，另外，在 OPC 作业运行的过程中，操作员提交的其他作业与这个时间的 OPC 作业产生也有可能产生资源冲突。对于第一种情况，可以在不影响业务的情况下调整作业顺序或者对于查询作业运用 DB2 的共享锁的技术，而第二种情况则要制定规范，规定在某时间断内不允许提交某些作业来解决。为了更好地开展系统分析工作，我将在以后的工作实践中不断地学习，提高自身素质和能力，为我国的软件事业贡献自己的微薄力量。

2024年11月论文1

云原生架构范文⁴²

2019年3月,我单位联合某高校研发了《程序在线评测比赛考试系统》。系统以程序代码在线提交自动评测功能为核心,分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等,支持对接教务平台。在项目中我担任系统架构师,负责架构设计工作。⁴³

本文以该系统为例,主要论述了云原生架构在项目中的具体应用。系统以 Spring Cloud 微服务框架开发,分为前端 Web 服务、平台保障服务、业务服务三部分。前端 Web 服务由负载均衡与服务器集群结合,实现高并发的前台界面;平台保障服务以 Eureka 为中心,由 API 网关、服务注册中心、监控平台等构成,实现基础服务框架;业务服务划分为多个微服务,基于 Docker 容器,协同工作实现具体业务功能。最终系统顺利上线,获得用户一致好评。⁴⁴

笔者在一个专为高校建设计算机专业智能教学一体化平台的单位任职,过往成果有《计算机组成原理仿真实验系统》等。2019年3月,我单位联合某大学研发了《程序在线评测比赛考试系统》项目(以下简称“OJ 系统”),以取代原有传统的编程上机考试平台。⁴⁵

系统以程序代码的在线提交自动评测功能为核心,主要分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等。题库模块主要负责试题和测试用例的管理,用户根据试题要求编写程序代码提交到系统,系统将测试用例与程序代码发送给评测机模块,由评测机自动编译、执行、判分,并将结果发送给其他相关模块进行统计;实验作业模块用于在线布置作业,从题库中选取试题,设置截止日期等要求;考试模块用于学生在线考试,按教师预先设置的参数自动从题库随机抽题生成试卷,以及向教务平台上传考试成绩;比赛模块主要用于 ACM 竞赛的培训;抄袭判定模块用于鉴定代码与他人代码雷同率;用户管理模块负责用户信息的管理。在这个项目中,我担任了系统架构师的职务,主要负责系统的架构设计相关工作。⁴⁶

2024年11月论文1

云原生架构以微服务和容器技术为代表,有服务化、强韧性、可观测性和自动化四类设计原则。通过服务化的设计原则,应用被分解为多个服务,可分别选择不同的技术,单个服务模块很容易开发、理解和维护,无需协调其他服务对本服务的影响;通过强韧性的设计原则,微服务可以分布式云化部署,负载均衡管理请求的分发,避免单机失败对整体服务的影响,以及弹性调整资源容量;通过可观测性的设计原则,能够对系统进行健康检查、指标监控、日志管理和链路追踪,提高系统运维、管理和排错能力;通过自动化的设计原则,可实现系统的自动化部署、自动化扩展伸缩、自动化运维、持续交付和集成,有效减少人工操作的工作量。⁴⁷

OJ 系统基于 Spring Cloud 微服务框架开发,将平台服务划分为三类,分别为前端 Web 服务、平台保障服务、业务服务。下面针对这三类服务展开具体说明。⁴⁸

1. 前端 Web 服务⁴⁹

前端 Web 服务主要提供给用户使用的界面,分为前置 Nginx 负载均衡服务器、前端网站 Nginx 集群。当用户通过网络访问系统时,首先会访问到前置的 Nginx 负载均衡服务器,负载均衡服务器会将请求转发到前端网站的 Nginx 集群,前端网站通过发起 http 请求来和后端交互,具体是通过 Ajax 方式来调用后端 REST API 接口。用户访问网站通过前置的 Nginx 负载均衡服务器来转发到前端网站集群,以起到将用户请求进行分流的作用。当前端网站集群中的部分服务发生故障时,系统仍可正常地对外提供服务。前置 Nginx 负载均衡服务器使用软件反向代理的方式来实现负载均衡,部署为路由模式,系统内部网络与外部网络分属于不同的逻辑网络,以实现系统内部与外部网络的隔离。在负载均衡算法的选择上,使用最小连接法,每当用户的请求来临时,任务分发单元会将任务平滑分配给最小连接数的前端网站节点,这样的架构以廉价且透明的方式扩展了服务器和网络的带宽,可以大大提升系统的并发量,同时保证网站前端整体的稳定性和可靠性。⁵⁰

