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。

■ 日 人取 Tray Lax I thick / HA / Hick □ NOI ○ NE 干 HORIC I MAINE

历年真题汇总分析

年份中	₩_~	运搬 工	试器三□	计区别对
202405	<u>论大数据</u> lambda 架 构⇔	论模型驱动架构设计 方法及其应用中	単元憲武は	论云上自动化坛维及 其应用↔
2023₽	可靠性分析与评价方 法=>	面向对象分析中	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。	<u>论湖仓</u> 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=2	论企业集成平台的理 解与应用=2	<u>论微服务</u> 架构及其应 用⇔
2020-3	论企业集成架构设计 及应用⇔	论软件测试中缺陷管 理及其应用↔	论云原生架构及其应 用↩	<u>论数据</u> 分片技术及其 应用⇔
2019+3	<u>论软件</u> 设计方法及其 应用↔	<u>论软件</u> 系统架构评估 及其应用⇔ ^Ⅰ	<u>论数据湖技术</u> 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⇨	<u>论软件</u> 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₽	<u>论软件</u> 系统建模方法 及其应用↔	<u>论软件</u> 架构风格↔	论无服务器架构及其 应用⇔	论软件质量保证及其 应用中
201643	<u>论软件</u> 系统架构评估中	<u>论软件</u> 设计模式及其 应用⇔	<u>论数据</u> 访问层设计技 术及其应用↔	<u>论微服务</u> 架构及其应 用⇔
2015↔	论应用服务器基础软 件↔	<u>论软件</u> 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?

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。

13

1.选择合适的项目

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)。↩

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
- (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。

包括内容: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部力学环境,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。←

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。

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 15³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

N

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laaS和PaaS中,从而减少业务代码开发人员的技术关注范围,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。

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,处理完成后云自动会关闭/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。

■ 8-9/2文章額 × +

变化通知等场景中。

2024年11月论文1

层次式架构范文。

摘要: 我所在的单位是国内主要的商业银行之一,作为单位的主要技术骨干,2019 年 1 月,我主持了远期结售汇系统的开发,该系统是我行综合业务系统的一个子系统,由于银行系统对安全性,可靠性,可用性和响应速度要求很高,我选择了三层 C/S 结构作为该系统的软件体系结构,在详细的设计三层结构的过程中,我采用了字符终端为表示层,CICS TRANSATION SERVER 为中间层,DB2 UDB 7.1 为数据库层,并采用了 CICS SWITCH组,并行批量的办法来解决设计中遇到的问题,保证了远期结售汇系统按计划完成并顺利投产,我设计的软件三层结构得到了同事和领导的一致认同和称赞。但是,我也看到在三层结构设计中存在一些不足之处,比如中间层的负载均衡算法过于简单,容易造成系统负荷不均衡,并行批量设计不够严谨,容易造成资源冲突等。 ←

我所在的单位是国内主要的商业银行之一。众所周知,银行的业务存在一个"二八定理",即银行的百分之八十的利润是由百分之二十的客户所创造。为了更好地服务大客户,适应我国对外贸易的 蓬勃发展态势,促进我国对外贸易的发展,2019年1月,我<u>行开展</u>了远期结售汇业务。 所谓的远期结售汇就是企业在取得中国外汇管理局的批准后,根据对外贸易的合同等凭证与银行制定合约,银行根据制定合约当天的外汇汇率,通过远期汇率公式,计算出交割当天的外汇汇率,并在那天以该汇率进行成交的外汇买卖业务。远期结售汇系统是我行综合业务系统的一个子系统,它主要包括了联机部分、批量部分、清算部分和通兑部分,具有协议管理、合约管理、报价管理、外汇敞口管理、帐务管理、数据拆分管理、报表管理、业务缩微和事后监督等功能。 我作为单位的主要技术骨干之一,主持并参与了远期结售汇系统的项目计划、需求分析、设计、编码和测试阶段的工作。中

由于银行系统对安全性,可靠性,可用性和响应速度要求很高,我选择了三层 C/S 结构作为该系统的软件体系结构,下面,我将分层次详细介绍三层 C/S 软件体系结构的设计过程。4

2024年11月论文1

- 1、表示层为字符终端。我行以前一直使用 IBM 的 VISUALGEN 2.0 附带的图形用户终端来开发终端程序,但在使用的过程中,分行的业务人员反映响应速度比较慢,特别是业务量比较大的时候,速度更是难以忍受。为此,我行最近自行开发了一套字符终端 CITE,它采用 VISUAL BASIC 作为开发语言,具有响应速度快,交互能力强,易学,<u>编码快</u>和功能强大的特点,在权衡了两者的优点和缺点之后,我决定选择字符终端 CITE 作为表示层。 \leftrightarrow
- 2、中间层为 CICS TRANSATION SERVER (CTS)。 首先, 我行与 IBM 公司一直保持着良好的合作关系,而我行的大部分技术和设备都采用了 IBM 公司的产品,其中包括了大型机,由于 CICS 在 IBM 的大型机上得到了广泛的应用,并在我行取得了很大的成功,为了保证与原来系统的兼容和互用性,我采用了 IBM 的 CTS 作为中间层,连接表示层和数据库层,简化系统的设计,使开发人员可以专注于表示逻辑和业务逻辑的开发工作,缩短了开发周期,减少开发费用和维护费用,提高了开发的成功率;其次,对于中间层的业务逻辑,我采用了我行一直使用的 VISUALAGE FOR JAVA 作为开发平台,它具有简单易用的特点,特别适合开发业务逻辑,可以使开发人员快速而准确地开发出业务逻辑,确保了远期结售汇系统的顺利完成;最后,由于采用了 CTS,确保了系统的开放性和互操作性,保证了与我行原来的联机系统和其他系统的兼容,保护了我行的原有投资。 ↔
- 3、数据层为 DB2 UDB7. 1. 由于 DB2 在大型事务处理系统中表现出色,我行一直使用 DB2 作为事务处理的数据库,并取得了很大的成功,在 DB2 数据库的使用方面积累了自己独到的经验和大量的人才,为了延续技术的连续性和保护原有投资,我选择了 DB2 UDB7. 1 作为数据层。 \leftrightarrow

但是,在设计的过程中我也遇到了一些困难,我主要采取了以下的办法来解决: 4

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,比如这次分配给鉴 N __CTS,下次则分配给鉴 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,我采用了当鉴 N __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。4

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,有一部分的帐<u>多</u>处理,比如一部分<u>内部户帐务</u>处理,或者代理收费业务和<u>总帐</u>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<u>行以前</u>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 PATITON 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。 \leftrightarrow

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<u>行以后</u>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<u>得不过够严</u>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。4

云原生架构范文。

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。↔

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 <u>微服务</u>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。4

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(以下简称为"0J系统"),以取代原有传统的编程上机考试平台。42

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。

2024年11月论文1

云原生架构以<u>微服务</u>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<u>本服务</u>的影响;通过强韧性的设计原则,<u>微服务</u>可以分布式<u>云化部</u> 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。↩

0J 系统基于 Spring Cloud <u>徵服务</u>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。↩

1. 前端 Web 服务↔

前端 Web 服务主要提供给用户使用的界而,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。如

平台保障服务用以实现后端微服务的基础框架,包括 API 路由网关、服务注册中心、服务监控组 件。API 网关收到前端的请求,不会直接调用后端的业务服务,而是首先会从服务注册中心根据当前 请求来获取对应的服务配置,随后通过服务配置再调用已注册的服务。当后端微服务存在多个实例时, 将采取负载均衡的方式调用。服务注册中心是整个云原生架构体系的核心部分,由 Spring Cloud 的 Eureka 组件来实现,专门提供微服务的服务注册和发现功能,涉及三种角色:服务提供者、服务消 费者和服务注册中心。API 路由网关、所有业务服务,以及服务监控平台组件都注册到服务注册中心。 通过服务注册中心两两互相注册、API 路由网关向服务注册中心注册多个实例等方式,来实现后端整 体服务的高可靠性。服务监控平台通过注册到服务注册中心、获取所有注册到服务注册中心的后端业 务服务,从而监控到所有后端业务服务的运行状态信息,最后收集并展示整个微服务系统的运行状态, 更进一步保证整个后端的服务质量。4

3. 业务服务↔

业务服务按功能模块,相应划分为题库服务、评测机服务、考试服务、比赛服务、抄袭判定服务 等。各服务单独打包,基于 Docker 容器,连同运行环境一起封装,根据实际情况可在一台或多台物 理机同时部署多个实例,服务启动后会将自身信息注册到服务注册中心。服务间协同工作,通过松耦 合的服务发现机制,动态调用对方 REST API 接口。对于压力较大的服务,如评测机服务、抄袭判定 服务等,将部署为多实例集群。以在线考试功能为例,用户进入考试时,考试服务核验考生信息通过, 调用腿库服务, 腿库服务返回试题, 由考试服务组合成试卷, 返回前端显示。用户交卷时, 提交的程 序代码到达考试服务,考试服务拆分后分发给题库服务,题库服务将程序代码和测试用例送入 MQ 队

列排队。然后由负载均衡机制,依次将队列中待评测程序分发给评测机服务编译、执行、判分,完成 评测后, 题库服务统计试题通过率, 考试服务统计成绩并向前端显示。在此期间服务请求者无需了解 其他服务对数据如何具体处理和分析。

2024年11月论文1

系统自 2019 年 10 月正式上线已运行一年有余,在学校的日常教学考试和竞赛培训中投入使用, 至今己有 3000 名以上的学生用户,评测了 70000 份以上的程序代码,获得了单位同事领导和学校教 师们的一致好评。在开发和试运行过程中,主要遇到了两个问题。一是跨域问题。0J 系统前后端分 离,前端通过 Ajax 访问后端服务。由于浏览器同源策略的限制,导致前端 UI 无法正常访问不同端口 和 IP 的后端服务。我们利用 Spring Boot 后端的 Cors 跨域机制解决了该问题。二是评测机宕机问题。 评测机服务需要执行用户提交的代码,部分用户短时间内提交了大量不安全代码,导致评测机集群中 所有实例全部 宣机。我们引入心跳机制、快照回滚机制,以及基于机器学习技术的预判断机制,使评 測机宕机时能够在10秒内自动重置恢复运行,最终解决了该问题。↩

实践证明, 0J 系统项目能够顺利上线, 并且稳定运行, 与系统采用了合适的架构设计密不可分。 经过这次云原生架构的方法和实施的效果后,我也看到了自己身上的不足之处,在未来还会不断更新 知识,完善本系统在各方面的设计,使整个 0J 系统能够更加好用,更有效地服务于高校师生。↔