

2019 年系统架构师考试科目三：论文

1. 论负载均衡技术在 Web 系统中的应用

负载均衡技术是提升 Web 系统性能的重要方法。利用负载均衡技术,可将负载(工作任务)进行平衡、分摊到多个操作单元上执行,从而协同完成工作任务,达到提升 Web 系统性能的目的。

请围绕“**负载均衡技术 Web 系统中的应用**”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,以及你在其中所承担的主要工作。

2.详细阐述常见的三种负载均衡算法,说明算法的基本原理。

3.详细说明你所参与的软件开发项目中,如何基于负载均衡算法实现 Web 应用系统的负载均衡。

【解析】

一、简要叙述所参与管理和开发的软件项目,需要明确指出在其中承担的主要任务和开展的主要工作。

二、现有的负载均衡算法主要分为静态和动态两类。静态负载均衡算法以固定的概率分配任务,不考虑服务器的状态信息,如轮转算法、随机法等;动态负载均衡算法以服务器的实时负载状态信息来决定任务的分配,如最小连接法等。

(1)轮询法。轮询法就是将用户的请求轮流分配给服务器,就像是挨个数数,轮流分配。这种算法比较简单,具有绝对均衡的优点,但是也正是因为绝对均衡,它必须付出很大的代价,例如它无法保证分配任务的合理性,无法根据服务器承受能力来分配任务。

(2)随机法。随机法是随机选择一台服务器来分配任务。它保证了请求的分散性达到了均衡的目的。同时它是没有状态的,不需要维持上次的选择状态和均衡因子。但是随着任务量的增大,它的效果趋向轮询后也会具有轮询法的部分缺点。

(3)最小连接法。

最小连接法将任务分配给此时具有最小连接数的节点,因此它是动态负载均衡算法。一个结点收到一个任务后连接数就会加 1,当结点发生故障时就将结点权值设置 0,不再给结点分配任务。最小连接法适用于各个结点处理的性能相似的情形。任务分发单元会将任务平滑分配给服务器。但当服务器性能差距较大时,就无法达到预期的效果。因为此时连接数并不能准确表明处理能力,连接数小而自身性能很差的服务器可能不及连接数大而自身性能极好的服务器。所以在这个时候就会导致任务无法准确地分配到剩余处理能力强的机器上。三、论文中需要结合项目实际工作,详细论述在项目中是如何基于负载均衡算法实现 Web 系统负载均衡的。

2. 论数据湖技术及其应用

近年来,随着移动互联网、物联网、工业互联网等技术的不断发展,企业级应用面临的数据规模不断增大,数据类型异常复杂。针对这一问题,业界提出“数据湖(DataLake)”这一新型的企业数据管理技术。数据湖是一个存储企业各种原始数据的大型仓库,支持对任意规模的结构化、半结构化和非结构化数据进行集中式存储,数据按照原有结构进行存储,无须进行结构化处理;数据湖中的数据可供存取、处理、分析及传输,支撑大数据处理、实时分析、机器学习、数据可视化等多种应用,最终支持企业的智能决策过程。

请围绕“数据湖技术及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细阐述数据湖技术，并从主要数据来源、数据模式(Schema)转换时机、数据存储成本、数据质量、面对用户和主要支撑应用类型等 5 个方面详细论述数据湖技术与数据仓库技术的差异。

3.详细说明你所参与的软件开发项目中，如何采用数据湖技术进行企业数据管理，并说明具体实施过程以及应用效果。

【解析】

一、概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

二、数据仓库是一个优化的数据库，用于分析来自事务系统和业务线应用程序的关系数据。数据仓库技术需要事先定义数据结构和数据模式(Schema)以优化快速 SQL 查询，其中结果通常用于操作报告和分析。数据经过了清理、丰富和转换，因此可以充当用户可信任的“单一信息源”。与数据仓库不同，数据湖能够同时存储来自业务线应用程序的关系数据，以及来自移动应用程序、物联网设备和社交媒体的非关系数据。在进行数据捕获时，无须定义数据结构或数据模式(Schema)。数据湖支持用户对数据使用不同类型的分析(如 SOL 查询、大数据分析、全文搜索、实时分析和机器学习等)，为企业智能决策提供支撑。

下面从主要数据来源、数据模式转换时机、数据存储成本、数据质量、面对用户和主要支撑应用类型等六个方面对数据湖技术和数据仓库技术进行比较：

特性	数据湖	数据仓库
主要数据来源	来自物联网设备、互联网、移动应用程序、社交媒体和企业应用程序的结构化、半结构化和非结构化	来自事务系统、运营数据库和业务线应用程序的结构化数据
数据模式转换时机	数据进入数据湖时不进行模式转换，在进行实际数据分析时才进行模式转换	在进入数据仓库之前（需要提前设计数据仓库的 Schema）
数据存储成本	通常基于非关系型数据库，数据存储成本相对较低	通常基于关系型数据库，数据存储成本高
数据质量	原始的、未经处理的数据	可作为重要事实依据的高质量数据
面对用户	业务分析师、应用开发人员和数据科学家	业务分析师
主要支撑应用类型	机器学习、预测分析、数据发现和分析	批处理报告、商务智能(BI)和数据可视化

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目是如何采用数据湖技术进行数据管理的，详细说明所采用的数据湖架构、主要的数据来源和质量、数据模式转换方式和时机、数据存储基础设施、系统主要用户和支撑的上层应用等，并对实际应用效果进行分析。

3. 软件系统架构评估及其应用

对于软件系统,尤其是大规模复杂软件系统而言,软件系统架构对于确保最终系统的质量具有十分重要的意义。在系统架构设计结束后,为保证架构设计的合理性、完整性和针对性,保证系统质量,降低成本及投资风险,需要对设计好的系统架构进行评估。架构评估是软件开发过程中的重要环节。

请围绕“软件系统架构评估及其应用”论题,依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目,以及你在其中所承担的主要工作。

2.详细阐述有哪些不同的软件系统架构评估方法,并从评估目标、质量属性和评估活动等方面论述其区别。

3.详细说明你所参与的软件开发项目中,使用了哪种评估方法,具体实施过程和效果如何。

【解析】

一、概要叙述你所参与管理或开发的软件项目,以及你在其中所承担的主要工作。

二、详细阐述有哪些不同的软件系统架构评估方法,并从评估目标、质量属性和评估活动等方面论述其区别。常见的系统体系架构分析方法有 SAAM 和 ATAM。

SAAM(Scenarios-basedArchitectureAnalysisMethod)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 的过程包括五个步要,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(ArchitectureTradeoffAnalysisMethod)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。

三、针对作者实际参与的软件系统架构评估工作,说明所采用的评估方法,并描述其具体实施过程和效果。

4. 论软件设计方法及其应用

软件设计(SoftwareDesign, SD)根据软件需求规格说明书设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及程序流程等,形成软件的具体设计方案。软件设计把许多事物和问题按不同的层次和角度进行抽象,将问题或事物进行模块化分解,以便更容易解决问题。分解得越细,模块数量也就越多,设计者需要考虑模块之间的耦合度。

请围绕“论软件设计方法及其应用”论题,依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细阐述有哪些不同的软件设计方法，并说明每种方法的适用场景。

3 详细说明你所参与的软件开发项目中，使用了哪种软件设计方法，具体实施效果如何。

【解析】

一、简要描述所参与管理和开发的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、详细阐述有哪些不同的软件设计方法，并说明每种方法的适用场景。

软件设计方法：

(1) 模型驱动设计。

模型驱动设计是一种系统设计方法，强调通过绘制图形化系统模型描述系统的技术和实现。通常从模型驱动分析中开发的逻辑模型导出系统设计模型，最终，系统设计模型将作为构造和实现新系统的蓝图。

(2) 结构化设计：

结构化设计是一种面向过程的系统设计技术，它将系统过程分解成一个容易实现和维护的计算机程序模块。把一个程序设计成一个自顶向下的模块层次，一个模块就是一组指令：一个程序片段、程序块、子程序或者子过程，这些模块自顶向下按照各种设计规则和设计指南进行开发，模块需要满足高度内聚和松散耦合的特征。

(3) 信息工程：

信息工程是一种用来计划、分析和设计信息系统的模型驱动的、以数据为中心的但对过程敏感的技术。信息工程模型是一些说明和同步系统的数据和过程的图形。信息工程的主要工具是数据模型图(物理实体关图)。

(4) 原型设计：

原型化方法是一种反复迭代过程，它需要设计人员和用户之间保持紧密的工作关系，通过构造一个预期系统的小规模的、不完整的但可工作的示例来与用户交互设计结果。原型设计方法鼓励并要求最终用户主动参与，这增加了最终用户对项目的信心和支持。原型更好地适应最终用户总是想改变想法的自然情况。原型是主动的模型，最终用户可以看到并与之交互。

(5) 面向对象设计：

面向对象设计是一种新的设计策略，用于精炼早期面向对象分析阶段确定的对象需求定义，并定义新的与设计相关的对象。面向对象设计是面向对象分析的延伸，有利于消除“数据”和“过程”的分离。

(6) 快速应用开发：

快速应用开发是一种系统设计方法，是各种结构化技术(特别是数据驱动的信息工程)与原型化技术和联合应用开发技术的结合，用以加速系统开发。快速应用开发要求反复地使用结构化技术和原型化技术来定义用户的需求并设计最终系统。

三、针对作者实际参与的软件系统开发项目，说明使用了哪种软件设计方法，并描述该方法实施后的实际应用效果。