

# 深圳技术大学本科毕业论文（设计）

## 开题报告

题 目	前端监控平台及其 SDK 的设计与实现				
学生姓名	王维	学号	202240292065	专业	计算机科学与技术
学 院	大数据与互联网学院	指导教师	郑俊虹		

### 本选题的意义及国内外发展状况：

#### 1. 前端监控的意义

前端监控在当今互联网时代的前端开发中扮演着至关重要的角色，其意义和重要性不容忽视。随着互联网应用的日益复杂和用户需求的不断增长，前端监控作为一种技术手段，为开发人员提供了关键的数据和洞察，有助于优化用户体验、提升应用性能、保障系统稳定性，进而实现持续的技术提升和业务发展。

首先，前端监控对于前端开发的意义在于实现对用户体验的全面把控。作为用户接触应用的第一入口，前端界面的流畅度、交互性以及加载速度直接影响用户对产品的感知。通过监控前端性能指标如页面加载时间、交互响应速度、错误率等，开发团队可以全面了解用户在不同环境下的真实体验，及时发现和解决存在的问题，从而提升用户满意度和忠诚度。

其次，前端监控在保障应用稳定性和可靠性方面发挥着关键作用。在大规模的用户访问下，应用往往面临着复杂多变的环境和多样化的设备，容易出现兼容性问题、性能瓶颈、甚至崩溃。通过监控系统的实时运行状态、错误日志、异常堆栈等信息，开发团队可以快速定位并解决问题，保障应用的稳定运行，避免因技术问题而影响用户体验和业务运营。

此外，前端监控也为持续优化和迭代提供了数据支持。通过收集用户行为数据、页面加载数据等，开发团队可以进行数据分析和挖掘，发现用户习惯、行为模式、热点等信息，为产品的优化和升级提供指导。基于监控数据分析，团队可以有针对性地进行功能改进、性能优化，不断提升产品质量和竞争力。

总的来说，前端监控对于前端开发具有重要意义。它不仅能够帮助开发团队及时发现和解决问题，提升用户体验和应用稳定性，更能够为持续优化和创新提供数据支持，推动产品不断发展。因此，在当前互联网发展的背景下，设计与实现一套高效可靠的前端监控平台及其 SDK 显得尤为重要，将有助于提升前端开发的效率和质量，为用户带来更优秀的用户体验。

#### 2. 国内发展状况及代表解决方案

技术应用多样性：中国的前端监控领域涌现了许多技术创新，包括自研的监控系统、开源监控工具等。例如，一些大型互联网公司自主研发了针对自身业务场景的监控平台，提供全面的性能指标监控和错误追踪。

数据驱动的用户体验优化：中国企业注重通过数据分析和监控指标优化用户体验，不断推出针对性的改进。大量用户行为数据的积累和分析，促使了一些公司采用个性化推荐、智能加载等技术手段优化前端性能。

行业标准化逐步成型：随着行业的发展，一些行业协会和标准化组织开始关注前端监控领域，制定了一些相关的技术标准，促进了行业内部的技术规范和共识。

ARMS

阿里云 ARMS (Application Real-Time Monitoring Service) 是阿里巴巴提供的一款全链路应用性能管理服务, 其中包括了前端监控系统。ARMS 前端监控系统主要针对 Web 应用的性能和用户体验进行监控和分析, 提供全面的性能数据和实时监控功能。可以帮助开发者实时监控应用的性能、识别问题并及时进行优化, 提升用户体验和应用的稳定性。

3. 国外发展状况及代表解决方案

开源社区和技术创新: 在国外, 开源社区的活跃程度较高, 很多前端监控工具和框架都源自于开源社区的共享与创新。例如, 一些监控平台的核心技术和算法经常在开源社区中得到积极讨论和贡献。

跨行业技术交流: 国外前端监控更加注重跨行业的技术交流和合作。不同行业的企业和技术团队倾向于分享经验和最佳实践, 推动了前端监控技术的跨界融合和创新。

用户隐私和数据安全: 在一些国外地区, 用户数据隐私保护意识较强, 监控平台在数据收集和处理方面更加注重用户隐私和数据安全, 对此类问题有更严格的规定和标准。

Sentry

Sentry 是一个开源的应用监控和错误追踪平台, 旨在帮助开发团队发现、诊断和解决软件中的异常和问题。它提供了广泛的错误追踪功能, 涵盖了前端、后端和移动端应用程序。

Sentry 支持多种主流编程语言和平台, 涵盖了广泛的应用类型。提供实时的错误监控和告警通知, 有助于快速响应和解决问题, 提供用户报告问题和用户体验数据收集的功能, 有助于全面了解用户需求和应用情况

综合来看, 无论国内外, 前端监控在不同的地区都处于快速发展的阶段。各地区在技术创新、数据应用和标准规范等方面存在一些差异, 但共同点在于都致力于提升用户体验、保障应用稳定性, 并且在技术创新和数据应用上持续探索前进。

研究内容:

1. 前端监控平台的设计与架构

需求分析与功能设计: 确定前端监控平台的基本功能, 包括性能监控、错误追踪、用户行为分析等, 结合实际需求设计功能模块。

技术选型: 从技术成熟度、可靠性和可扩展性方面, 选择最合适技术、工具和框架。

架构设计与扩展性考量: 设计灵活的架构, 考虑平台的可扩展性和兼容性, 使其能适应不同规模和类型的前端应用。

2. 数据分析与可视化

数据存储与处理: 设计合理的数据存储方案, 支持大规模数据的存储和处理, 确保监控数据的安全性和可靠性。

数据分析与可视化: 利用数据分析技术, 将监控数据转化为可视化的报表和图表, 提供直观的数据展示和分析功能。

3. 测试与评估

功能测试与性能评估: 对设计的前端监控平台和 SDK 进行全面的测试和性能评估, 确保其稳定性和可靠性。

实际案例应用: 在真实的前端应用中进行测试和应用实践, 收集反馈并根据实际情况进行优化和改进。

#### 4. 前端监控 SDK 的设计与架构

微内核架构：也称插件化架构，插件就是将一些特性或能力添加到已有程序中的程序，插件往往需要遵循主体程序制定的一些规范，也只能运行在指定的主体程序中。采用微内核架构可以增强可扩展性，保持内核稳定，降低程序复杂性。

发布/订阅模式：一种常见的软件架构模式，用于解耦多个组件或对象之间的通信。这种模式基于一种消息传递机制，允许不同的组件在不直接互相交互的情况下进行通信。

错误采集：对 JS 代码运行错误，异步错误，接口请求错误，静态资源加载错误的监控与采集

性能数据采集：利用浏览器提供的 performance api 采集 FP、FCP、LCP、CLS、TTFB 等性能指标。

用户行为采集：收集用户在浏览页面的过程中的连续行为，比如鼠标点击，资源加载，接口调用，代码报错，路由变化

数据上报：将各模块产生的数据以合理的数据结构组织并通过某些手段上报给服务器

#### 研究方法、手段及步骤：

##### 1. 研究方法

###### 1) 文献综述与调研

通过文献综述和调研，梳理相关领域的已有研究成果、行业报告和技术文档，了解前端监控平台的基本原理、发展历程、应用场景和技术特点。这个步骤对于建立研究背景、问题定义和方法选择非常重要。

###### 2) 平台调研与比较分析

选择代表性的前端监控平台，如 New Relic、Datadog、Sentry 等，并对它们进行深入调研和比较分析。评估这些平台的功能特点、性能指标监控、用户体验、数据输出质量、成本等方面。可以通过使用体验、功能对比、案例研究等方法进行评估。

###### 3) 实证研究与案例分析

选择具有代表性的应用案例，针对不同前端监控平台进行实证研究和案例分析。通过部署监控方案、收集数据、分析监控结果，并基于实际数据来评估各个平台在性能、用户体验和问题诊断方面的表现。这可以通过定量数据分析和可视化来展现。

###### 4) 调查问卷与用户反馈

设计调查问卷，邀请前端开发人员、运维工程师等相关人群，了解他们对不同前端监控平台的使用体验、优缺点的认知。收集用户反馈和意见，有助于补充实证研究的数据，增加研究的全面性。

###### 5) 最佳实践总结与建议提炼

基于文献综述、调研、实证研究和用户反馈，总结不同前端监控平台的优缺点，归纳出最佳实践和建议。这些建议可以涵盖平台选择、配置优化、数据解读、问题定位与优化策略等方面，为开发人员提供实际可行的指导。

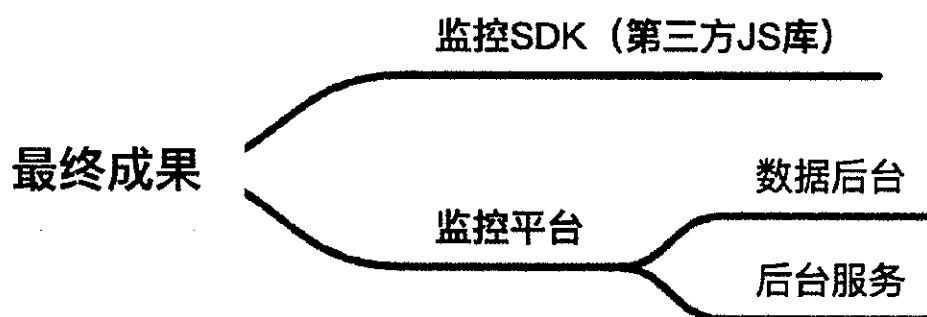
##### 2. 研究手段：

1) 查阅相关文献资料并了解所有现有平台的具体功能与相关实现

2) 进行需求分析，包括功能性需求与非功能性需求

3) 根据需求进行技术选型，侧重点放在复杂度低，易用上

- 4) 对整体项目进行排期，包括前期调研，中期开发，后期论文总结，保质保量。
  - 5) 搭建相应的开发环境，后端基于 Node 开发，前端基于 React。
  - 6) 对项目进行评估测试，对存在的问题或者不好的地方进行改进优化
3. 研究步骤：
- 1) 文献综述和平台初筛  
查阅学术文献、技术博客和行业报告，了解前端监控平台的基本原理和发展历程。选择数个代表性的前端监控平台，初步评估其特点和适用场景，筛选出符合研究目标的平台。
  - 2) 案例选择和设计  
根据不同应用类型和特点，选择适合的应用作为案例，确保代表性和多样性。在选定的前端监控平台上，设计监控方案和指标体系，确保能够全面评估性能和用户体验。
  - 3) 数据采集和分析  
在所选应用上部署监控方案，收集数据并建立数据存储和分析系统。运用统计方法和可视化工具对收集的数据进行分析，识别性能瓶颈和用户行为模式。  
结果总结和实践建议
  - 4) 结论与展望  
总结研究的主要发现和结论，回答研究问题。探讨前端监控平台发展的趋势，可能的改进方向和未来研究的方向。
4. 面临的挑战
- 1) 在 SPA 页面盛行的时代，让原本已被 W3C 规范化的首屏时间失去了意义，需要一种新的算法来测量页面首屏时间。
  - 2) 前端不断发展，让 PV 的计算成为了难题，比如，未关闭的浏览器 tab 几小时之后再次浏览，该不该再计一次 PV？商品列表页，看完一屏之后，向上滚动会再加载新的一屏，PV 该算一次还是多次？
  - 3) SDK 会采集各种各样的数据，数据采集的时机不一致，要求的上报时机可能也不一致，协调采集与上报的时机关系到数据的正确性。
  - 4) 用户行为还原是高效解决问题的基础，主要手段有记录用户行为栈，用户行为录屏，但录屏的传输压力和服务器压力是必须考虑的，所以要设计一套完善的策略。权衡功能与成本。
5. 最终成果/预期目标：



**参考文献：**

[1] 徐江伟. Web 前端实时异常监控与报警平台的设计与实现 [D]. 湖北: 华中科技大学

学, 2016.

[2] 杨晨. Web 前端异常响应监控与 WRG 生成系统的设计与实现[D]. 四川: 电子科技大学, 2023.

[3] 张钊源, 刘晓瑜, 鞠玉霞. Node.js 后端技术初探[J]. 中小企业管理与科技, 2020(22): 193-194.

[4] 何伟, 王浩, 蔡忠平. 基于 Kafka 的公交 GPS 数据实时通讯系统的设计与应用[J]. 城市公共交通, 2022, 288(6): 34-37, 41.

[5] 王水波. 基于 Web 前端数据可视化研究与应用[D]. 陕西: 西安电子科技大学, 2017.

[6] 校丽丽. 基于 MVVM 模式的前端框架插件库设计与实现[D]. 四川: 西南科技大学, 2019.

[7] 唐斌斌, 叶奕. Vuejs 在前端开发应用中的性能影响研究[J]. 电子制作, 2020(10): 49-50, 59.

[8] 杨明. Web 前端性能优化方法研究分析[J]. 信息记录材料, 2023, 24(8): 161-163.

[9] 戴志诚, 程劲草. 基于虚拟 DOM 的 Web 前端性能优化研究[J]. 计算机应用与软件, 2017, 34(12): 21-25, 31.

[10] 王兵兵, 陈能, 丁雅雪. 基于 MySQL 的智慧社区助老关爱平台数据库设计[J]. 信息通信, 2019(2): 152-154.

[11] 柴青山. 基于 Node.js 的新华社大屏幕蓝信监控系统应用研究[J]. 中国传媒科技. 2023(10)

[12] 郭春霞. 基于 Spark 的 Web 应用前端性能监控系统的设计与实现[D]. 北京交通大学. 2023

[13] 马特·弗里斯比. JavaScript 高级程序设计[M]. 人民邮电出版社. 2019

[14] 郑黎黎, 方菽兰, 许德鹏. 互联网中基于用户行为的 Web 异常检测系统研究[J]. 技术与市场. 2022, 29(02)

[15] 周沐玲. 高并发访问下的移动 Web 前端浏览性能优化研究[J]. 重庆科技学院学报(自然科学版). 2023, 25(05)

[16] 闵亮, 薛格格, 张玉欣. 基于 Node.JS 博客系统的设计与实现[J]. 电子设计工程. 2022. 30(07)

学生签名: 王雅

2023 年 12 月 6 日

指导教师意见：

该生已对毕业设计内容进行充分调研。  
同意开题。

签名：郑俊虹

学院领导意见：

同意。

签名：傅向华  
2023年12月08日