**XXX有限公司**

**外网门户网站系统适配改造方案**

**万户网络**

2022年1月

**目录**

[1. 项目分析 1](#_Toc31738)

[1.1. 项目背景 1](#_Toc21410)

[1.2. 重点事项分析 1](#_Toc28766)

[1.2.1. 功能迁移内容 1](#_Toc19836)

[1.2.2. 基金超市适配 2](#_Toc223)

[1.2.3. 适老化改造 2](#_Toc6259)

[1.2.4. 数据迁移内容 2](#_Toc18096)

[1.2.5. 性能调优要点 2](#_Toc32381)

[1.2.6. 备份策略 2](#_Toc27451)

[2. 项目建设目标 4](#_Toc5502)

[3. 技术可行性分析 5](#_Toc32368)

[3.1. 信创替代可行性分析 5](#_Toc8409)

[3.2. 用信创迁移可行性分析 5](#_Toc9356)

[3.3. 数据迁移可行性分析 5](#_Toc4253)

[3.4. 信息安全的可行性分析 5](#_Toc32543)

[4. 整体方案 7](#_Toc4836)

[4.1. 设计原则 7](#_Toc4967)

[4.2. 技术路线 8](#_Toc4446)

[4.2.1. 面向服务构架（SOA） 8](#_Toc16424)

[4.2.2. B/S结构 8](#_Toc1067)

[4.2.3. 采用J2EE开放标准的技术路线 8](#_Toc10161)

[4.2.4. 基于组件技术开发 10](#_Toc3169)

[4.2.5. 安全技术 11](#_Toc22197)

[4.2.6. 利用Web Services技术优化系统集成和二次开发架构 11](#_Toc7164)

[4.2.7. 支持XML技术作为系统集成和数据传递接口标准 12](#_Toc30482)

[4.2.8. 数据库缓存服务 13](#_Toc24066)

[4.3. 架构设计 14](#_Toc5575)

[4.3.1. 系统总体架构 14](#_Toc26920)

[4.3.2. 系统部署架构 15](#_Toc32167)

[4.3.2.1. 部署方式1（满足系统正常运行） 15](#_Toc22584)

[4.3.2.2. 部署方式2（高可用模式） 15](#_Toc29830)

[4.4. 适配设计 16](#_Toc32519)

[4.4.1. 技术选型原则 16](#_Toc22263)

[4.4.2. 技术路线选型 17](#_Toc26557)

[4.4.3. 技术适配架构 17](#_Toc11990)

[4.4.4. 服务器端适配 17](#_Toc25533)

[4.4.4.1. 服务器适配 17](#_Toc809)

[4.4.4.2. 数据库适配 17](#_Toc5232)

[4.4.4.3. 中间件适配 18](#_Toc21546)

[4.4.4.4. 操作系统适配 18](#_Toc13244)

[4.4.5. 终端适配 18](#_Toc3188)

[4.4.5.1. 浏览器适配 18](#_Toc24823)

[4.4.5.2. 操作系统适配 18](#_Toc14493)

[4.4.5.3. 流式软件适配 18](#_Toc7818)

[4.4.5.4. 版式软件适配 19](#_Toc8406)

[4.5. 系统适配 19](#_Toc19880)

[4.5.1. 外网门户网站适配 19](#_Toc31807)

[4.5.1.1. 前端适配 19](#_Toc8389)

[4.5.1.2. 功能适配 19](#_Toc29705)

[4.5.1.2.1. 系统管理 19](#_Toc682)

[4.5.1.2.2. 功能模型 23](#_Toc7340)

[4.5.1.2.3. 内容协同 25](#_Toc8036)

[4.5.1.2.4. 权限管理 29](#_Toc32179)

[4.5.1.2.5. 网站监测 31](#_Toc25943)

[4.5.1.2.6. 数据中心 33](#_Toc17636)

[4.5.2. 基金超市适配 35](#_Toc18495)

[4.5.2.1. 内容架构适配 35](#_Toc26589)

[4.5.2.2. 前端页面适配 35](#_Toc7397)

[4.5.2.3. 功能适配 35](#_Toc20310)

[4.5.2.4. 数据接口适配 37](#_Toc23506)

[4.5.3. 适老化建设及适配 37](#_Toc16806)

[4.5.3.1. 建设依据 37](#_Toc19014)

[4.5.3.2. 功能详情 37](#_Toc11838)

[4.6. 适配验证 44](#_Toc17469)

[4.6.1. 适配验证规划 44](#_Toc19000)

[4.6.2. 国产计算机整机验证 45](#_Toc30540)

[4.6.3. 国产基础软件适配 46](#_Toc11825)

[4.6.4. 应用软件适配验证 46](#_Toc7568)

[4.6.4.1. 整体性验证 46](#_Toc26620)

[4.6.4.2. 兼容性验证 46](#_Toc273)

[4.6.4.3. 性能验证 46](#_Toc18166)

[4.6.4.4. 业务流畅性验证 46](#_Toc4934)

[4.6.4.5. 数据迁移验证 47](#_Toc8202)

[4.6.4.6. 接口可用性验证 47](#_Toc406)

[4.7. 优化策略 47](#_Toc28312)

[4.7.1. 部署优化 47](#_Toc14095)

[4.7.2. 缓存优化 47](#_Toc8676)

[4.7.2.1. 全局缓存 47](#_Toc31914)

[4.7.2.2. 局部缓存 48](#_Toc8889)

[4.7.3. 设计优化 48](#_Toc13523)

[4.7.3.1. 合并与压缩前台js,css等静态文件 48](#_Toc15203)

[4.7.3.2. 异步处理 48](#_Toc21518)

[4.7.3.3. 动态页面静态化 48](#_Toc7209)

[4.7.4. 数据库优化 48](#_Toc2035)

[4.7.4.1. 加索引优化SQL查询速度 48](#_Toc18673)

[4.7.4.2. 换查询条件优化SQL查询速度 48](#_Toc18134)

[4.7.4.3. 不利用数据库生成主键 49](#_Toc24760)

[4.7.4.4. 采用PreparedStatement方式 49](#_Toc5586)

[4.7.4.5. 优化代码，减少数据库查询次数 49](#_Toc30201)

[4.8. 系统测试 49](#_Toc18)

[4.8.1. 准备工作 49](#_Toc10056)

[4.8.1.1. 系统基础功能验证 49](#_Toc20884)

[4.8.1.2. 测试团队组建 49](#_Toc24988)

[4.8.1.3. 工具的选择 49](#_Toc30894)

[4.8.1.4. 预先的业务场景分析 50](#_Toc16934)

[4.8.2. 测试计划 50](#_Toc19749)

[4.8.2.1. 性能测试领域分析 50](#_Toc31654)

[4.8.2.2. 确定性能目标 50](#_Toc12439)

[4.8.2.3. 制定测试计划的实施时间 50](#_Toc922)

[4.8.3. 测试脚本设计与开发 50](#_Toc23421)

[4.8.3.1. 测试环境设计 51](#_Toc11712)

[4.8.3.2. 测试场景设计 51](#_Toc799)

[4.8.3.3. 测试用例设计 51](#_Toc20668)

[4.8.3.4. 脚本和辅助工具的开发及使用 52](#_Toc10275)

[4.8.4. 测试执行与管理 52](#_Toc9447)

[4.8.4.1. 建立测试环境 52](#_Toc6819)

[4.8.4.2. 执行测试脚本 52](#_Toc22154)

[4.8.4.3. 测试结果记录 52](#_Toc26608)

[4.8.5. 测试分析 53](#_Toc20252)

[4.8.5.1. 测试环境的系统性能分析 53](#_Toc18283)

[4.8.5.2. 硬件设备对系统性能表现的影响分析 53](#_Toc5432)

[4.8.5.3. 其他影响因素分析 53](#_Toc1814)

[4.8.5.4. 测试中发现的问题 53](#_Toc24925)

[5. 迁移方案 54](#_Toc15825)

[5.1. 数据迁移方案 54](#_Toc13767)

[5.1.1. 规划迁移工作 54](#_Toc30817)

[5.1.2. 数据表结构及附件迁移 54](#_Toc31189)

[5.1.3. 数据迁移应急策略 66](#_Toc32678)

[5.1.3.1. 系统迁移实施前的异常 66](#_Toc24025)

[5.1.3.2. 系统迁移实施过程中的异常 67](#_Toc2492)

[5.1.3.3. 系统迁移实施后的异常 67](#_Toc12880)

[5.1.4. 数据迁移工具 67](#_Toc29905)

[5.1.4.1. 数据抽取与数据加载功能 67](#_Toc30148)

[5.1.4.2. 信息传输功能 71](#_Toc25078)

[5.1.4.3. 数据转换功能 74](#_Toc14065)

[5.1.4.4. 作业流程工具 81](#_Toc30521)

[5.1.4.5. 系统监控功能 83](#_Toc5349)

[5.1.4.6. 数据迁移步骤 84](#_Toc3014)

[5.1.5. 数据迁移风险及应对方式 85](#_Toc13607)

[5.2. 应用迁移方案 86](#_Toc21990)

[5.2.1. 功能说明 86](#_Toc2103)

[5.2.2. 系统部署 86](#_Toc12813)

[5.2.3. 迁移实施 87](#_Toc14045)

[5.2.4. 注意事项 87](#_Toc4963)

[5.3. 迁移工作计划 87](#_Toc10019)

[6. 上线方案 88](#_Toc29020)

[6.1. 上线运行策略 88](#_Toc15857)

[6.2. 上线切换实施 88](#_Toc25446)

[6.2.1. 调研评估 88](#_Toc27290)

[6.2.2. 切换时间 88](#_Toc25270)

[6.2.3. 上线前工作内容 88](#_Toc13288)

[6.2.4. 系统切换 88](#_Toc11186)

[6.2.5. 上线后维护 89](#_Toc31330)

[7. 信创技术与软硬件配置建议 90](#_Toc17303)

[7.1. 目前应用厂商可支持的所有技术路线 90](#_Toc480)

[7.2. 软硬件资源配置建议 92](#_Toc3133)

[7.2.1. 服务器配置建议清单 92](#_Toc9389)

[7.2.2. 软件配置建议清单 93](#_Toc24372)

[7.2.3. 负载均衡建议清单（软负载/可选） 93](#_Toc25797)

[7.2.4. 服务器操作系统分区建议 93](#_Toc12942)

[8. 建设预算 94](#_Toc29680)

[8.1. 软件费用 94](#_Toc7429)

# 技术可行性分析

## 信创替代可行性分析

外网门户系统产品可通过升级的方式完成信创项目替代。系统支持主流的信创软硬件环境，并在数据迁移、安全管理、性能优化、智能组件、运维平台等各方面都有对应的解决方案与技术支持。

## 用信创迁移可行性分析

信创改造内容包含：服务器、中间件、数据库、流式文件、版式文件，系统均已具备直接迁移至信创环境下的能力，迁移后基本不影响用户体验。

## 数据迁移可行性分析

应用厂商已完成数据库适配工作，并提供一整套数据迁移方案，确保历史数据完整迁移至信创环境下。

## 信息安全的可行性分析

经调研信息安全可行，能满足相关政策文件的信息安全要求。参考文件如下：

* 《关于信息安全等级保护工作的实施意见》(公通字[2004]66号)
* 《信息安全等级保护管理办法》(公通字[2007]43号)
* 《信息安全技术网络安全等级保护基本要求》（GBT 22239-2019）
* 《信息安全技术网络安全等级保护测评要求》（GBT 28448-2019）
* 《信息安全技术 网络安全等级保护安全设计技术要求》（GBT 25070-2019）
* 《信息系统密码应用基本要求》（GM/T0054-2018）

开发过程中在不引入新的安全风险基础上，需要按照现有的应用系统安全要求来开发。该项目原则上需遵循XXX有限公司各系统安全、易用、稳定等相关要求。需要关注移动类应用接口安全；总体而言安全需求有足够的技术支持实现，其中：

* 软件安全方面：应注意必要的软件授权设计、配置文件安全、输入输出验证、异常处理安全、会话安全、文件安全、参数操作安全等方面的内容；
* 人员身份与访问控制安全方面：整体建议加入身份认证系统用户认证平台与统一账户权限平台，符合我司对于账户权限的安全设计，同时注意身份管理、权限管理、访问控制及日志审计等安全功能；
* 数据安全方面：应注意必要的数据分类、加密等安全内容，针对处理关键敏感数据（客户数据、密码密钥等）应注意进行加密存储及传输；
* 主机网络配置安全方面：应注意应用系统主机所在网络安全域及主机操作系统、数据库、中间件的安全配置管理。

# 整体方案

## 设计原则

系统的设计紧密结合实际应用总体建设原则，同时根据项目自身特点，采取如下设计原则：

（1）先进性原则

在系统设计时，将充分应用先进和成熟的技术，满足建设的要求，把科学的管理理念和先进的技术手段紧密结合起来，设计出先进合理的业务流程。系统使用先进成熟的技术手段和标准化产品，在满足实用性的基础上，充分预见未来技术发展趋势，保证系统在不替换现有设备、不损失前期投资的情况下能方便地升级和扩容，最大可能地延长系统的整体生命周期，确保系统能在未来较长时间内充分发挥作用。

（2）安全性原则

系统从网络安全、系统安全、代码安全三方面来保证系统的安全运行。在网络安全与系统安全得以保证的前提下，系统在代码全安上进行加固。代码安全上系统应该规避现有已知的JAVAWEB安全漏洞。

（3）实用性原则

各系统系统依据市场主流的系统功能设计 ，紧密结合实际情况进行设计开发，确保系统实用、高效和方便。

（4）稳定性原则

系统建设与产品选型严格遵循国际国内有关标准，充分考虑技术和产品的成熟性，优先使用已在实际中获得规模化运用的技术和产品；采取模块化、分布式技术构建系统，同时采用具有高可用性的软件使系统具备冗余备份和快速恢复能力，分散故障风险，降低软件系统故障概率，提高系统的总体可靠性。

（5）扩展性原则

遵循标准化准则，采用开放性好、标准化强的技术建设系统，保障系统易于扩展业务流程和不断丰富其功能。数据库结构应充分考虑未来的发展和移植，具有良好的扩展性和伸缩性。

## 技术路线

### 面向服务构架（SOA）

SOA（Services-Oriented Architecture，SOA）是“面向服务的架构”，运用这个架构使得着眼于应用程序提供的服务，而不是着眼于提供服务所需的组件。SOA使得应用系统的设计清晰化而且促进组件的重用。应用系统中所有的接口的定义与信息模型――包括数据及其语义、对象与过程模型――都高度一致。

通过建立SOA架构，将对本项目的各业务系统，不论是旧的或新的，都能够通过服务的包装，成为随用即取的IT 资产，以服务的形式对外发布，以松耦合原则实现共享，并可将各种服务快速整合，开发出组合式应用，达到“整合即开发”的目的，实现对各业务需求的快速响应。

采用SOA架构具有以下主要优势：

为用户或其他应用系统访问本系统的业务逻辑提供的访问形式为服务（Service）

独立于用户接口（user interfaces）定义服务接口（service interfaces）

采用标准的形式发布，以便于其他用户或应用系统发现和调用这些服务

服务接口可以从目录中查获，并能动态捆绑不同的实现

允许采用不同类型的编码/数据模型

### B/S结构

本系统开发采用B/S结构（Browser/Server，浏览器/服务器模式）。B/S结构是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。浏览器通过Web Server 同数据库进行数据交互。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。

B/S结构的维护和升级方式简单。当前，软件系统的改进和升级越发频繁，B/S架构的产品明显体现着更为方便的特性。B/S架构软件的维护重点在服务器，所有的客户端都是基于浏览器，基本不需要做任何的维护。针对用户规模大，分支机构多的特点，任何维护升级的工作只需要针对服务器进行，提高工作效率和运维成本。

### 采用J2EE开放标准的技术路线

本项目的建设需要在总体结构设计上充分考虑整体性、可用性、稳定性、成熟性、灵活性和开放性的基本技术要求，要体现出安全性、可扩展性、可管理性、用户界面友好性和高性能等特点。通过采用基于J2EE标准的应用服务器中间件作为应用系统的开发架构，以实现应用系统的快速开发、灵活部署。故本方案技术路线是采用开放的、标准的、主流的、成熟的J2EE系统平台、开发手段与信息技术规范。

采用基于行业标准或得到广泛使用的事实上的行业标准的技术和架构，有利于降低技术风险和对特定供应商的依赖性；采用的开放系统架构，有利于保持系统的向后兼容性、可集成性和可扩展性。

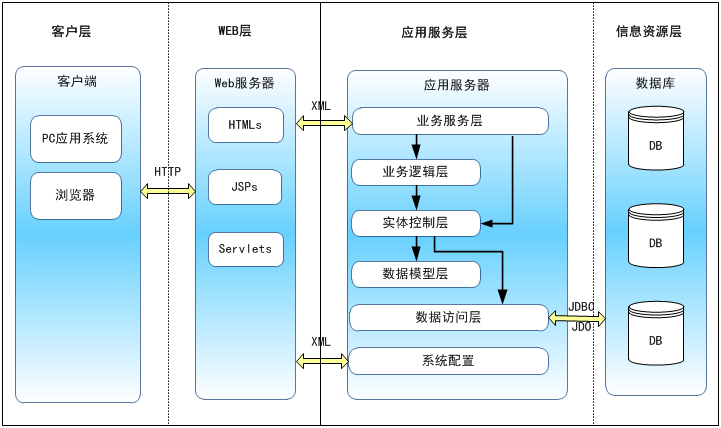
本项目的设计应充分考虑“标准和开放”的原则，要支持各种相应的软硬件接口，使之具有灵活性和扩展性，具备与多种系统互连互通的特性，在结构上实现真正开放。在系统建设中广泛采用遵循国际国内标准的系统和产品以便于系统的互联和扩展，综合反映在可移植性、互操作性和集成性上。

J2EE平台由一整套服务（Services）、应用程序接口（APIs）和协议构成，它对开发基于Web的多层应用提供了功能支持，包括JDBC、JNDI、EJB、XML、JMS、JTA等13种技术规范。

J2EE是使用Java技术构建大型应用系统的标准平台与环境，是开发应用的一种事实上的工业标准，利用Java 平台可以简化应用系统解决方案的开发、部署和管理相关等复杂问题。J2EE技术的基础是核心Java平台或Java 2平台的标准版，J2EE不仅巩固了标准版中的许多优点，例如“编写一次、随处运行”的特性、方便存取数据库的JDBC API技术以及能够在网络应用中保护数据的安全模式等等，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlets API、JSP（Java Server Pages）以及XML技术的全面支持。

J2EE体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，J2EE降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成强有力支持。J2EE为搭建具有可伸缩性、灵活性、易维护性的应用系统提供了良好的机制，具有保留现存IT资产、高效开发、支持异构环境、稳定的可用性等优势。

J2EE使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据他们所在的层分布在不同的机器上。J2EE 的多层应用模型能够为不同的每种服务提供一个独立的层，下图是 J2EE 典型的四层结构：



（1）客户层：客户层为用户提供交互界面，侧重于用户图形界面（GUI），用于完成接收用户的输入，并向应用程序服务器发出处理请求和显示返回的处理结果。客户层选用浏览器和专用客户端作为本系统的客户端。

（2）WEB层：Web层的主要功能是同客户交户，向用户提供服务，主要功能是提供HTML与基于类似html页面语法规则的标签语言的模板页面，接受用户的请求，调用业务功能并把结果返回到客户端，此层没有业务逻辑的处理，而只是调用应用服务层面提供的服务。Web层由Web服务器和Web组件组成。Web组件包括Servlets、HTMLs。

（3）应用服务层（业务逻辑层）：应用服务层是应用的核心，主要任务是处理某个特定业务领域的业务逻辑。业务代码由运行在业务层的enterprise beans来执行，包括SessionBeans、EntityBeans两种类型的服务器端组件，应用服务层经常被称作Enterprise JavaBeans（EJB）层。应用服务层和Web层一起构成了3层J2EE应用的中间层，而其他两层是客户层和信息系统层。

（4）信息资源层：信息资源层运行信息系统软件，这层包括基础设施系统，应用服务层通过JDBC、JCA 这些 J2EE 的标准和针对不同应用的JCA 来访问后台的信息系统。

### 基于组件技术开发

组件化是应用的一个整体趋势。由于组件所具有预制性、封装性、透明性、互操作性、通用性的特征，这使得“Write once，Run everywhere”成为技术上的可能，单位也更多地接受标准的、互换（Interchangeable）、互操作（Interoperable）的成品软件包（Commercial Off-the-Shelf）的单位业务组件以快速地组装成新的系统应用。

采用组件技术可以实现灵活的接口定义、执行代码运行时刻的联编/载入以及通讯网络协议，支持异构分布应用程序间的互操作性及独立于平台和编程语言的组件重用。基于组件，具有平台无关性程序的编写十分简单，因为业务逻辑被封装成可复用的组件，开发者可以集中精力解决手头的业务问题。

在本项目应用的成熟组件，主要包括两个方面：

1）J2EE服务器所提供的内在支持，包括安全、事务管理、JNDI(Java Naming and Directory Interface)寻址、远程连接、生命周期管理等服务；

2）应支撑平台提供的日志服务、报表服务、交换服务、数据访问服务、检索服务、认证服务等公共组件。

基于组件的开发是普通应用程序开发过程的改进，它具有如下特点：

应用程序由各自独立的组件组成，这些组件的开发和部署保持相对的独立性，而且很可能是由不同的团队开发和部署；

通过仅对这种应用程序的某些组件进行升级，从而对其进行小幅度的升级；

组件可以在不同应用程序之间共享，因此可对它们复用，但同时也产生了项目之间的依赖关系；

尽管并非与基于组件完全密不可分，但基于组件的应用程序倾向于分布式结构；

建设多种信息技术渠道的解决方案，多渠道共享业务逻辑；

分层是从逻辑上将子系统划分成许多集合，而层间关系的形成要遵循一定的规则。通过分层可以降低子系统间的依赖关系，使系统以更松散的方式耦合，从而更易于建设、维护和进化；可以根据业务的变化，通过快速开发或者配置调整系统以适应系统新的业务需求。

### 安全技术

采用身份认证服务体系，部署应用级身份认证网关，利用身份认证载体，进行用户身份认证，实现用户的身份鉴别、访问控制、三员分立以及数据库安全保障。

### 利用Web Services技术优化系统集成和二次开发架构

Web Service（Web服务)是一种分布式的计算技术，通过标准的XML协议和信息格式来发布和访问商业应用服务，它是SOA的具体实现的一种手段。

Web Service具备如下特点：

1）互操作性：任何的Web服务都可以与其他Web服务进行交互。因为SOAP协议是所有供应商都支持的标准，避免了在各种协议之间转换带来的麻烦。

2）普遍性：Web服务使用HTTP协议和XML进行通信。因此，任何支持这些技术的设备都可以拥有和访问Web服务。

3）易用性：Web 服务的概念易于理解，任何开发语言都可以用来编写Web服务。目前已经有许多工具可以开发和部署Web服务，还有一些工具可以将JavaBeans、Enterprise JavaBeans部署为Web服务。

Web Service技术及其相关技术体系，包括XML、SOAP、WSDL、UDDI等。Web Service是一种新的Web应用程序分支，他们是自包含、自描述、模块化的应用，可以发布、定位、通过Web调用。一旦部署以后，其他Web Service应用程序可以发现并调用它部署的服务。从而方便的被其它部门所使用，以便应用到跨部门的业务中去。如下图所示：



### 支持XML技术作为系统集成和数据传递接口标准

XML（Extensible Markup Language）定义了一种跨越任何平台、任何语言的数据表示格式，在XML所传输的信息中除包括用户数据外，还包括对数据格式的描述信息，任何接收端的XML都能够方便地“解码”，并将XML标准格式的数据转换成本地数据。这样便可实现不同硬件平台、不同操作系统平台、不同开发语言的应用之间平滑地通信。

XML即可扩展标记语言，它与HTML一样，都是SGML(Standard Generalized Markup Language，标准通用标记语言)。XML是跨平台的，依赖于内容的技术，是当前处理结构化文档信息的有力工具。扩展标记语言XML是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然XML占用的空间比二进制数据占用更多的空间，但XML极其简单易于掌握和使用。

XML与数据库不同，数据库提供了更强有力的数据存储和分析能力，例如：数据索引、排序、查找、相关一致性等，XML仅仅是展示数据。事实上XML与其他数据表现形式最大的不同是：他极其简单，XML的简单使其易于在任何应用程序中读写数据。

### 数据库缓存服务

对于高并发高访问的Web应用程序来说，数据库存取瓶颈是个很大的问题。为了减少数据库读取负担，提高数据读取速度，我们可以在web/app层与db层之间加一层cache层。cache存取的媒介是内存，而一台服务器的内存容量一般都是有限制的，不像硬盘容量可以做到TB级别。所以，考虑采用分布式的cache层，这样更易于破除内存容量的限制，同时又增加了灵活性。

（1）缓存原理：

在app层和db层之间加入cache层，每个app服务器都可以绑定一个Cache服务器，每次数据的读取都可以从Cache服务器中取得，如果没有，再从db层读取。而当数据要进行更新时，除了要发送update的sql给db层，同时也要将更新的数据发给Cache服务器。

（2）Memcached缓存技术：

使用较为成熟的第三方解决方案最大的优势就在于在节省自身研发成本的同时，大大降低实施风险，本方案采用目前比较流行且成熟的开源第三方 Cache产品基于对象的分布式内存 Cache 软件 Memcached。

Memcached在Web2.0社区非常流行，主要是因为以下几个原因：

通信协议简单，API接口清晰；

高效的 Cache 算法，基于 libevent 的事件处理机制，性能卓越；

面向对象的特性，对应用开发人员来说非常友好；

所有数据都存放于内存中，数据访问高效；

作为一个 Cache 软件，那么更多的是需要通过应用程序来维护 Memcached 中的数据与数据库中数据的同步更新。这时候的 Memcached 基本可以理解为比数据库更为前端的一个 Cache 层。

## 架构设计

### 系统总体架构

系统的系统逻辑结构按照 Java EE 应用标准模型分为多层结构，即客户端表示层、业务层逻辑层（可细分为服务器应用逻辑层、数据访问层）。

在系统应用逻辑架构中，客户端表现层可以是任何浏览器工具；业务层使用

SpringBoot/SpringMVC技术实现，简单成熟，而且性能的可调用性非常强；服务器应用逻辑层基于SOA架构。

系统采用多层应用模式架构模型进行结构设计，并可以通过WebServices技术，方便内外数据调用，将来可以与其他业务系统建立数据标准接口，实现该系统与其它不同业务系统数据的调用和处理，可以轻松的在该平台上整合部署。

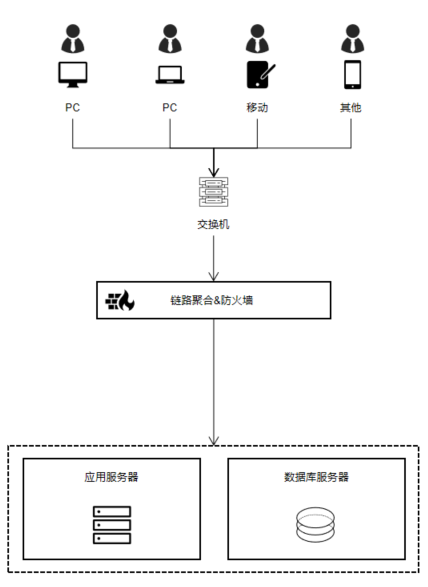


（图：系统架构）

### 系统部署架构

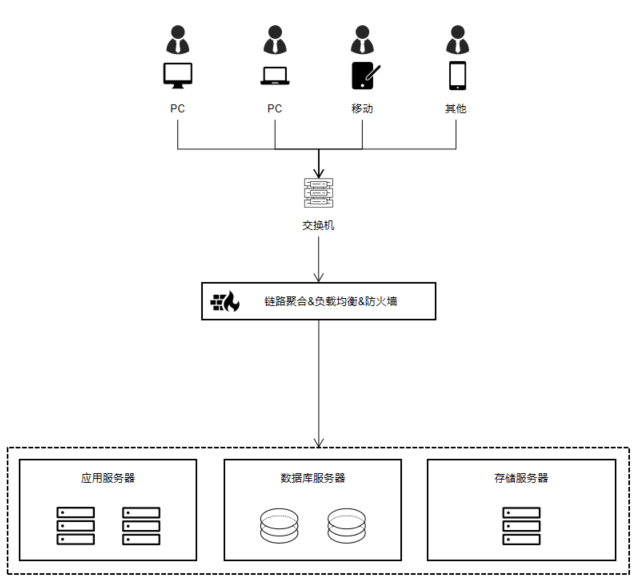
#### 部署方式1（满足系统正常运行）

此部署方式基于满足系统日常基础运行考虑，采用一台应用服务器和一台数据库服务器来实现系统的正常运行，系统部署架构图如下：



#### 部署方式2（高可用模式）

此部署方式基于满足系统高可用模式，通过两台应用服务器、两台数据库服务器加一台存储服务器，同时使用负载均衡软件、数据库高可用软件保障系统的高可用性。系统部署架构图如下：



## 适配设计

### 技术选型原则

* 安全可靠，自主可控

选用完全自主知识产权，自主设计的CPU核心技术。其他关键软硬件产品要与CPU完全适配，拥有自主知识产权并具备安全可靠的设计与生产能力。

* 满足需要，能用可用
* 核心软硬件产品组合要具备良好的兼容性、稳定性与性能，保证系统整体IT特性良好。同时，应用系统须能满足用户业务需求与使用需求，确保当前可用、够用。
* 持续发展，迭代优化

具备围绕指令集构建的软件生态、指令集的实现方法和自由扩展指令集的权力。在技术路线的整体发展上，能够引导自身产业生态发展的能力。

### 技术路线选型

考虑安全可靠生态的稳定性和长期服务保障需求等因素，优先选择具备本地化服务及核心层面技术支撑能力的技术路线选型。优先选择在我省具有适配中心，并有芯片、操作系统、API软件三个核心层面的技术服务团队在本地提供即时性服务保障和兜底性技术支持的单位。

考虑到项目实施的长期性和延续性，优先选择自主程度较高的技术路线。

### 技术适配架构

本次项目国产化适配总体架构主要包括以下几个方面：硬件环境改造、软件环境改造、办公套件替换、应用系统改造、客户终端替换5个方面，部署方案如下图所示：

### 服务器端适配

#### 服务器适配

已实现国产服务器适配，包括鲲鹏、龙芯、飞腾等国产芯片服务器环境，兼容文件转换、国产化服务器、国产化中间件、国产化操作系统、国产化版式文件、版式软件等。

#### 数据库适配

已实现国产数据库适配，包括达梦、人大金仓等数据库，用于数据安全存储和管理的数据库产品。主要功能包括：数据存储、访问控制、身份鉴别、安全审计和数备份恢复等功能。

产品部署在国产服务器，以后台服务形式运行，数据库管理员及用户在管理主机上通过图形化管理工具或命令行工具，可实现对数据库对象（表、视图、约束、索引、触发器、存储过程等）的配置管理；开发人员可通过标准化数据库访问接口开发基于安全数据库的应用系统和软件产品。

#### 中间件适配

已适配国产中间件，包括东方通、金蝶等中间件，用于为上层应用提供运行环境，实现对上层应用的部署和动态管理。主要功能包括Web容器、EJB容器、数据源服务、集群管理等。

产品部署在国产服务器，以后台服务形式运行，用户通过管理控制台或者命令行工具完成上层应用的部署、启动、停止等操作。

#### 操作系统适配

已实现国产操作系统适配，包括中标麒麟、银河麒麟、统信OS等。

提供中文化的操作系统环境和常用图形管理工具。支持多种安装方式，提供完善的系统服务和网络服务；集成多种易用的编译器并支持众多开发语言；提供掉电保护和备份恢复机制。

### 终端适配

#### 浏览器适配

已实现国产浏览器适配，包括UOS浏览器、360安全浏览器等，全面支持中标麒麟、银河麒麟、中科方德、深度、红旗、UOS、普华、一铭等国产操作系统，以及龙芯、兆芯、飞腾、海光、鲲鹏等国产芯片，支持高清视频在线播放，有效满足国产计算平台办公需求，并致力于打造国产操作系统统一规范的WEB应用平台，提升基于浏览器的WEB生态体系的建设速度。

#### 操作系统适配

已实现国产操作系统适配，包括中科方德、银河麒麟等操作系统，支持在国产化硬件环境正常、稳定运行；兼容了国产化浏览器、流式软件、版式软件等。

#### 流式软件适配

已实现国产流式软件适配，包括金山WPS及永中办公软件，用于处理文字、表格及幻灯片的流式办公软件。

#### 版式软件适配

已实现国产版式文件适配，包括福晰、数科版式软件用于处理OFD格式文件的版式办公软件。主要功能包括:阅读、编辑、保存、打印。

## 系统适配

### 外网门户网站适配

#### 前端适配

保留已完成的外网门户网站前端设计，直接迁移至信创环境内；通过对管理端及用户端的兼容性适配，保障管理端及用户端的正常访问。

#### 功能适配

将已完成的外网门户网站管理产品迁移适配至新的信创环境内，包括但不限于以下内容：

本系统共分为8个功能模块：站群管理、功能模型、内容协同、权限管理、网站检测、数据分析、会员中心和两微一端。以下是系统功能模块图：



##### 系统管理

站群管理包含发布点管理，工作流，站点配置等与站点相关的信息在此模块维护。所有站点使用一个系统统一管理，各站点之间具有独立的域名，站点SEO信息，站点栏目和风格模板，每个站点可以指定不同的管理员，让不同的人员负责管理发布站点数据。

**发布点管理**

发布点的设计是为实现站点分离部署功能而设计，发布点即前台站点部署位置，站点新建和编辑时，可以选择发布点，如果不选择发布点，站点与后管平台在同一服务器上且在同一应用程序根目录下，选择了发布点系统将会把站点的静态资源文件同步到发布点服务器上。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 1.2 | 功能名称 | 发布点管理 |
| 功能描述 | 发布点设计主要是为站点前台提供服务器部署和存储方式。不同站点可以发布到不同的发布点，也可以发布到同一个发布。发布点类型有本地，FTP。系统会通过不同的发布点类型以不同的传输协议将站点的前台主题样式文件和模板文件以及静态生成的网页文件同步到站点绑定的发布点。 | | |
| 数据描述 | 【发布点表】（whir\_dev\_publishpoint） | | |
| 业务约束 | 如果是FTP类型的发布点,需要预先在指定的发布服务器上预先配置FTP服务。 | | |
| 设计约束 | 发布点类型分为本地和FTP，当发布点类型为本地时，发布路径必填且唯一，当发布点类型为FTP时，发布点IP、用户名、密码、端口号必填，发布点IP与用户名组合唯一。 | | |
| 数据校验 | 发布点名称、发布点类型必填 | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**站点配置**

站点配置是站点信息的扩展维护，主要用来存储站点前台的相关信息包括站点logo、站点水印设置、站点SEO信息、站点版权信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 1.3 | 功能名称 | 站点配置 |
| 功能描述 | 站点配置表与站点表一对一的关系，站点配置表中的siteid字段与站点表id字段做主外键关联，一个站点只存在一条站点配置信息  站点配置表维护的信息有：网站LOGO、归档图片、归档图片左边距、归档图片上边距、是否开启水印、水印类型（图片、文字）、水印字体样式、水印图片、水印文字、水印位置、版权信息、备案信息、SEO标题、SEO关键词、SEO描述。 | | |
| 数据描述 | 【站点配置表】（whir\_dev\_site\_deploy） | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | -无- | | |
| 异常处理 | 保存失败，不存入数据库 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**工作流**

每个站点具有自己的工作流配置。工作流设计是为信息发布提供审核功能。系统不限制创建工作流个数，每个工作流下可管理工作流节点，节点个数不作限制

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 1.4 | 功能名称 | 工作流 |
| 功能描述 | 工作流具备的功能有：工作流创建、工作流编辑、工作流删除、工作流回收站、工作流节点管理、工作流节点新增、工作流节点排序、工作流节点编辑、工作流节点删除、工作流节点回收站。 | | |
| 数据描述 | 【工作流表】（whir\_dev\_workflow） 【工作流节点表】（whir\_dev\_workflow\_node）。表whir\_dev\_workflow\_node的workflow\_id 字段与whir\_dev\_workflow的id 做主外键关联。  工作流表whir\_dev\_workflow 的siteid 与站点表的id 做主外键关联 | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | 工作流名称唯一，同一个工作流下节点名唯一 | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**站点主题**

为了方便开发人员或是维护人员对项目交付后的维护工作，系统设计提供出站点主题的维护管理。站点主题包含了站点模板，站点样式。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 1.5 | 功能名称 | 站点主题 |
| 功能描述 | 站点主题功能：文件夹列表，文件列表，下载、编辑、新建、重命名、搜索、删除、同步资源（同步样式和图片文件到发布点）、同步模板 | | |
| 数据描述 | 文件IO操作 | | |
| 业务约束 | 站点需要设定主题，如果未设置将无法呈现主题文件夹里的文件夹和文件 | | |
| 设计约束 | -无- | | |
| 数据校验 | 创建模板时，文件名格式是否符合文件名格式要求，文件名是否存在 | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**文件管理**

文件管理是管理当前站点下的所有附件包含图片、文件、音频、视频

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 1.6 | 功能名称 | 文件管理 |
| 功能描述 | 主要功能：列表，下载、批量下载、打水印、删除、批量水印、搜索 | | |
| 数据描述 | 【附件表】（whir\_dev\_upload\_file） 表字段siteid与站点表的id 做主外键关联。 | | |
| 业务约束 | 文件删除时，只删除数据表whir\_dev\_upload\_file中的数据，不进行文件删除处理 | | |
| 设计约束 | -无- | | |
| 数据校验 | -无- | | |
| 异常处理 | 操作失败，显示失败信息 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

##### 功能模型

功能模型是信息发布管理的核心模块，系统预设9大基础模型。后管平台的管理员可以通过功能模型的表单和表单字段功能快速变更信息维护表单的字段属性。预设的9大基础模型有：单页图文、新闻信息、列表信息、公共类别、公共表单、问卷调查、在线投票、信息报送、互动信箱。功能模型的管理主要是针对这9大基础模型的扩展。用户可在9大基础模型上扩展新的功能模型，创建新的表单，字段和实体数据表。

模型管理

模型管理实现了在原有9大基础模型上进行模型扩展，主要解决数据分散存储和功能的快速扩展。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 2.1 | 功能名称 | 模型管理 |
| 功能描述 | 模型管理功能：列表、新增、编辑、删除、回收站、字段管理、表单管理 | | |
| 数据描述 | 【模型表】（whir\_dev\_model） | | |
| 业务约束 | 创建模型数据时，需要同步创建模型实体表并初始化模型字段，并且创建默认表单 | | |
| 设计约束 | -无- | | |
| 数据校验 | 基础模型必选、模型名称和实体表名必填、模型表名唯一 | | |
| 异常处理 | 保存失败，不存入数据库，具不创建实体表 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**表单管理**

表单是一组表单输入项的集合，用于文章发布时组合配置需要输入的表单项数据，表单管理实现了在同一个功能模型下，同一张数据表里根据不同的栏目需求设计出不同的表单输入组合。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 2.2 | 功能名称 | 表单管理 |
| 功能描述 | 功能：列表、新增、编辑、删除、回收站、字段管理、复制 | | |
| 数据描述 | 【表单表】（whir\_dev\_form）表字段model\_id与模型表whir\_dev\_model里的id字段主外键关联 | | |
| 业务约束 | 表单的创建需要基于模型来创建 | | |
| 设计约束 | 表单的管理从模型管理里单个模型信息进入 | | |
| 数据校验 | 表单名称唯一、同一个模型下只能存在一个默认的表单 | | |
| 异常处理 | 保存失败，不存入或更新数据库 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**字段管理**

字段管理里存储的是功能模型里所有的字段，字段数据应与实体表字段一一对应。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 2.3 | 功能名称 | 字段管理 |
| 功能描述 | 功能：列表、新增、编辑、删除、回收站、排序，字段新增成功后，实体数据表创建同字段名的实体字段。 | | |
| 数据描述 | 【字段表】（whir\_dev\_form）表字段model\_id与模型表whir\_dev\_model里的id字段主外键关联 | | |
| 业务约束 | 字段的创建需要基于模型来创建 | | |
| 设计约束 | 字段的管理从模型管理里单个模型信息进入 | | |
| 数据校验 | 字段名称唯一、字段类型必选 | | |
| 异常处理 | 保存失败，不存入或更新数据库 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**表单字段管理**

表单字段实际是对字段进行表单归类，同一个字段可以在多个表单中，字段类型和字段名称不变但是字段的别名等其他信息可以在不同表单中设置不同的值。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 2.4 | 功能名称 | 表单字段管理 |
| 功能描述 | 功能：列表、新增、编辑、删除、回收站、排序，字段新增成功后，实体数据表创建同字段名的实体字段。 | | |
| 数据描述 | 【表单字段表】（whir\_dev\_form\_field）表字段field\_id与字段表whir\_dev\_field里的id字段主外键关联  表 whir\_dev\_form\_field 字段form\_id 与表单表whir\_dev\_form里id字段做主外键有关联 | | |
| 业务约束 | 字段的创建需要基于表单来创建 | | |
| 设计约束 | 字段的管理从表单管理里单个表单信息进入 | | |
| 数据校验 | 字段名称在当前模型中唯一、字段类型必选 | | |
| 异常处理 | 保存失败，不存入或更新数据库 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

##### 内容协同

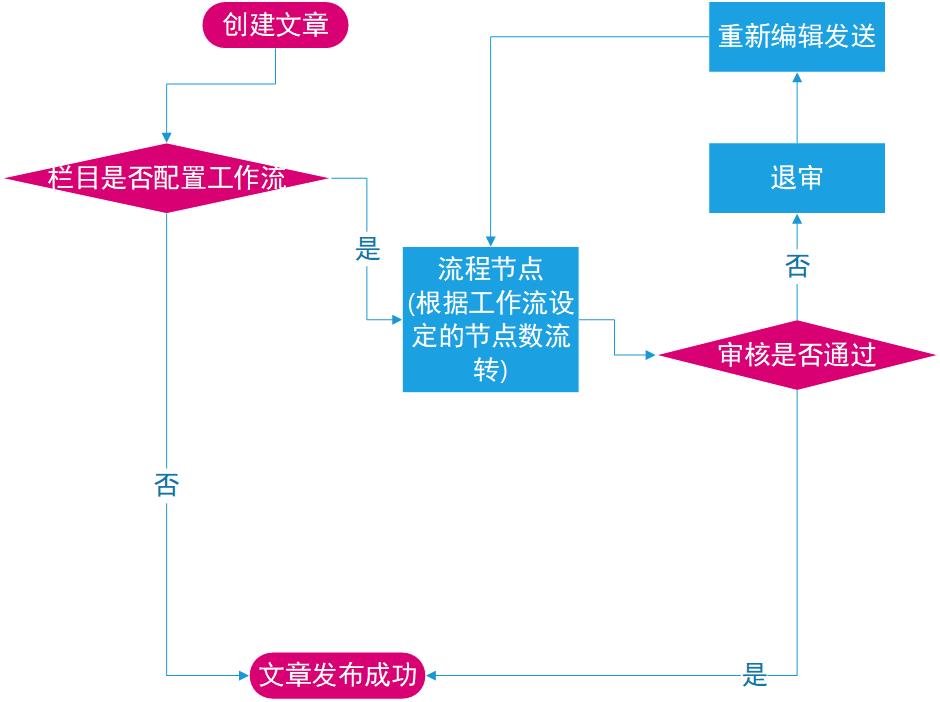
**栏目管理**

栏目管理管理站点前台栏目以及一些辅助性的信息栏目，系统栏目设计采用树型无限层级方式来呈现栏目结构关系。栏目分为三种栏目：1、普通栏目；2、专题栏目；3、菜单栏目；普通栏目与专题栏目都归属于站点。菜单栏目是菜单功能型栏目，栏目本身不归属于任何站点，但是栏目的数据按站点区分。

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 3.1 | 功能名称 | 栏目管理 |
| 功能描述 | UI展示形式：树型结构  功能：列表、回收站、新增、编辑、删除、排序、复制、快速创建、禁用、启用、附属栏目管理。  栏目通过设置功能模型来实现栏目的信息维护功能，通过绑定模板将栏目数据在前台页面呈现 | | |
| 数据描述 | 【栏目表】（whir\_dev\_column）表字段model\_id与模型表whir\_dev\_model里的id字段主外键关联，字段form\_id 与表单表whir\_dev\_form 字段id做主外键关联，字段siteid 与站点表whir\_dev\_site字段id做主外键关联;字段sub\_category\_id与专题类型表whir\_dev\_sub\_categor字段id 做主外键关联，sub\_category\_id不为0表示此栏目为专题栏目 | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | 栏目名称、功能模型、表单必填，栏目目录唯一 | | |
| 异常处理 | 保存失败，不存入或更新数据库 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**内容管理**

内容管理是站点栏目文章信息的维护管理模型，站点文章信息通过栏目进行分类，文章信息通用功能操作有：列表、回收站、流程中、预发布、新增、编辑、删除、历史、推送、移动、映射、还原、审核、排序、导入、导出。列表与回收站的呈现方式依据不同的功能模型呈现方式不同。列表呈现的字段与搜索条件字段可以通过自定义的配置来呈现。文章审核机制，根据栏目选择的工作流进行层级审核流转。文章审核业务流程如下：



**专题管理**

专题是某事件或主题的信息内容的聚合。专题中可能存在信息列表，图片特效，视频、音频播放、表单提交等多种表现形式。 一个好的专题管理系统要求制作效率高，时效性强，内容高聚合，互动性强以及出色的页面设计。

为了能实现快速搭建专题信息，专题管理设计了专题类别，通过专题类别来快速创建同等栏目结构和相同页面表现形式的专题。

专题管理包含专题类别管理、专题类别栏目管理、专题管理、专题栏目管理4个子功能。专题类别实现相同页面表现形式和栏目结构的专题模板与栏目的管理。专题管理维护所有的专题信息，专题栏目管理实现在相同类别下的专题的个性化，如：栏目个数的变动，栏目名称的个性化。

**生成发布**

本系统设计了动态与静态发布两种形式的前台页面发布，动态页面实现后管平台数据与前台数据呈现的实时更新，即后台发布文章，管理员无需做任何其他操作，刷新前台页面即可看到最新发布的文章。静态页面是为了缓存服务器压力和数据库的访问压力，提升网站访问吞吐量而设计的，静态页面需要通过后台操作来生成，系统设计了静态页面的同步生成、增量生成与全量生成。

同步生成：文章详细内容页，在新增编辑时同步生成，并同步生成指定自动生成的栏目首页，和绑定生成的其他页面。

增量生成：在生成管理界面设计了可以按站点，按栏目、按专题、按文章创建的时间段来生成相应的静态页面

全量生成：在生成管理界面，用户可以通过选择站点整站生成来实现前台静态页面的全量生成。

**表单提交**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 3.5 | 功能名称 | 表单提交 |
| 功能描述 | 表单提交的设计是为开发人员或运维人员可以通过后台调表单提交项目来实现表单的字段和样式的变更。  表单提交需要与栏目进行绑定，处理表单提交接口通过绑定的栏目找到对应的物理表，将数据保存到物理表。提交后的数据管理员用户可以在内容管理绑定的栏目下查看到，并进行相应的维护。 | | |
| 数据描述 | 【表单提交表】（whir\_dev\_sendform） 表中字段columnid 与栏目表whir\_dev\_column 的字段id做主外键关联 | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | 表单名称，栏目必填 | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

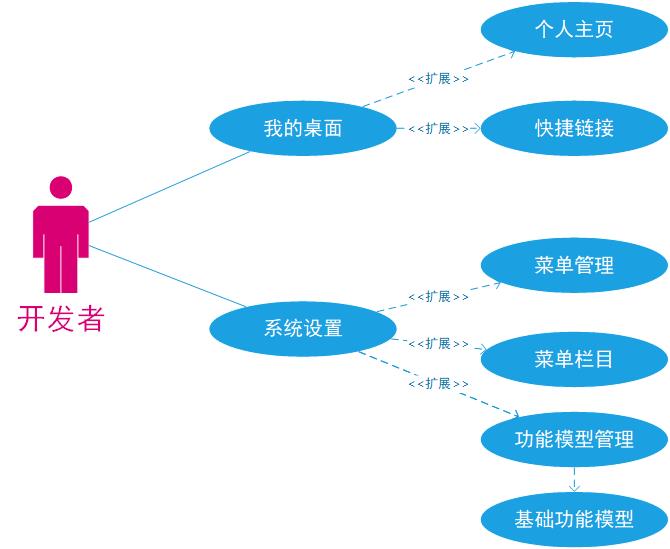
**全文检索**

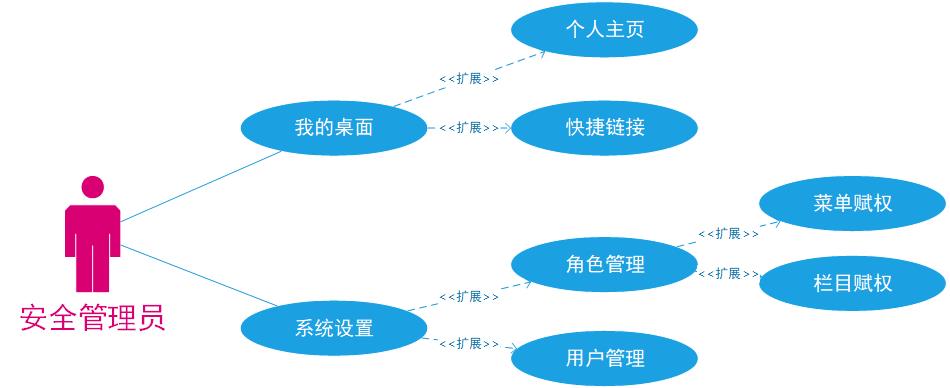
系统基于Solr+IK中文分词实现网站信息的全文检索。Solr是一个独立的企业级搜索应用服务器，它对外提供类似于Web-service的API接口。系统通过http请求，向搜索引擎服务器提交一定格式的XML文件，生成索引；也可以通过Http Get操作提出查找请求，并得到XML/Json格式的返回结果。在设计中，将站点，专题，栏目，标题，内容，创建时间等作为索引。信息在创建和编辑时会通过消息中间件异步更新索引。

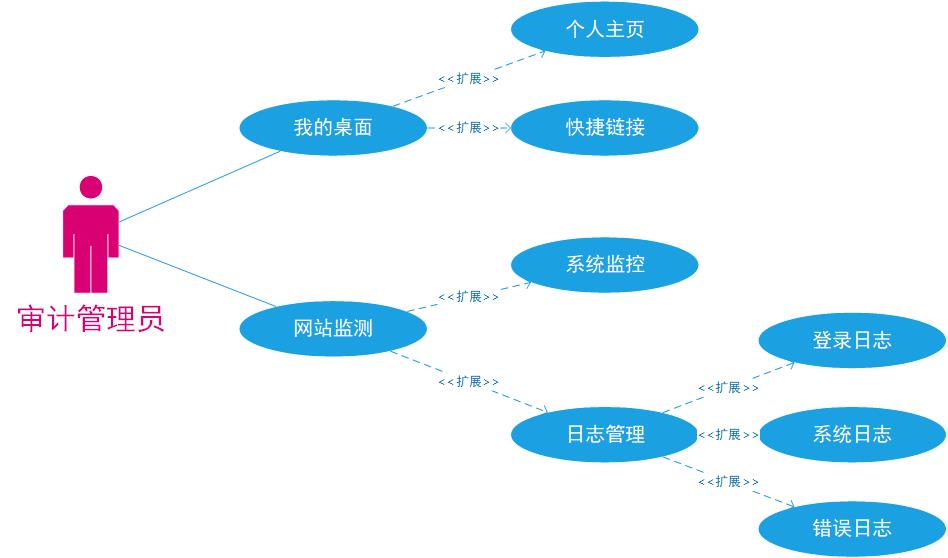
##### 权限管理

系统的权限分为站点权限，菜单权限和栏目权限。用户的权限信息由角色来承载。系统的角色主要分为5种：系统管理员、开发者、安全管理员、审计管理员、普通管理员。开发者、安全管理员、审核管理员的权限是系统设计之初就已固定。系统设计可以配置是否开启三员权限管理。如果未开启三员权限管理，系统管理员的权限几乎是最大的权限。如果开启系统管理员将无审计权限和权限分配权限。普通管理员的权限是由系统管理员或安全管理员配置。普通管理员的权限不能大于系统管理员或对应的上级角色权限。

角色用例图如下：







系统管理员在开启最大权限时，拥有系统全部功能和操作，关闭最大权限时，系统管理员拥有除开发者、安全管理员、审核管理员以外的全部功能和操作，同时还具备通用功能我的桌面的所有的功能。

**角色管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 4.1 | 功能名称 | 角色管理 |
| 功能描述 | 角色管理维护整个系统的角色信息。具有权限的用户可创建角色，给角色赋予站点的管理员权限（可配置一个或多个）。角色具备站点权限后，可以配置对应站点的菜单权限的栏目权限以及专题栏目权限。角色的赋权不能越权，即子角色的权限不能大于父角色的权限。 | | |
| 数据描述 | 【角色表】（whir\_dev\_role） 【角色权限表】（whir\_dev\_role\_power）。表whir\_dev\_role\_power的role\_id 字段与whir\_dev\_role的id 做主外键关联。 | | |
| 业务约束 | -无- | | |
| 设计约束 | 创建角色时，必须要设置可管理的站点。否则角色在菜单赋权和菜单赋权时无法操作。除普通管理员外的其他类型的管理员不可通过后管平台创建。 | | |
| 数据校验 | 角色名称唯一，上级角色必选。 | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**用户管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 4.2 | 功能名称 | 用户管理 |
| 功能描述 | 用户管理维护可操作后管平台的用户。用户管理解决的是系统的认证。用户必须指定角色才具备系统的功能权限操作。根据三级等要求，系统设计了用户的密码的高强度验证。 | | |
| 数据描述 | 【用户表】（whir\_dev\_user）。表whir\_dev\_user的role\_id字段与whir\_dev\_role 的id做主外键关联。 | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | 用户名符合正则验证要求，且唯一，密码符合正则验证要求，并进行加密码。 | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

##### 网站监测

**信息预警**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 5.1 | 功能名称 | 信息预警 |
| 功能描述 | 信息预警分为2个子功能：  信息预警设置：树型显示站点所有栏目，设置发布周期（可选：每天、每周、每月、每年、自定义天数内）、设置要求发布的数量。  预警信息：利用定时任务调度框实现每日执行查询变存储未及时更新的栏目及未更新条数和已更新条数。 | | |
| 数据描述 | 【预警设置表】（whir\_dev\_warning）、【预警信息表】（whir\_dev\_warning\_log） | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | -无- | | |
| 异常处理 | 数据查询失败，返回状态码500 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**易错词管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 5.2 | 功能名称 | 错别字检测 |
| 功能描述 | 易错词管理：设置常用的易错词汇，字段有易错词、正确词。操作功能：新增、编辑、删除  易错词检测：内容管理发布或编辑文章出，当输入项失去焦点时进行内容检测，如果内容包含易错词则给出易错词提示，并阻止内容的提交。 | | |
| 数据描述 | 【易错词表】（whir\_dev\_typos） | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | 易错词新增时，易错词与正确词必填 | | |
| 异常处理 | 数据提交失败，返回状态码500 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**日志管理**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 5.5 | 功能名称 | 日志管理 |
| 功能描述 | 系统分为三种日志监控：  系统日志：管理员操作后管的记录日志  登录日志：管理员登录到后管的记录日志  错误日志：系统后管运行过程中产生的运行异常信息。  系统通过AOP设计思想，在不影响系统正常操作和运行的情况下实时记录日志信息。  日志记录的信息包括日志标题，日志时间，操作者IP，日志详细内容（系统日志与登录日志存储操作过程中发送的请求数据，异常日志记录详细的异常信息，所属栏目，日志发生时间，日志产生的URL） | | |
| 数据描述 | 【系统日志表】whir\_dev\_system\_log  【登录日志表】whir\_dev\_login\_log  【异常日志表】whir\_dev\_exception\_log | | |
| 业务约束 | -无- | | |
| 设计约束 | -无- | | |
| 数据校验 | -无- | | |
| 异常处理 | -无- | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

##### 数据中心

**发布统计**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 6.1 | 功能名称 | 发布统计 |
| 功能描述 | 从4个方面进行统计  按栏目统计：以栏目列表显示每个栏目在指定范围内的总发布数，待审数，文章总数  按角色统计：以角色列表显示每个栏目在指定范围内的总发布数，待审数，文章总数  按机构统计：以机构列表显示每个栏目在指定范围内的总发布数，待审数，文章总数  按用户统计：以用户列表显示每个栏目在指定范围内的总发布数，待审数，文章总数 | | |
| 数据描述 | -无- | | |
| 业务约束 | 未开启工作流的栏目待审数总为0，发布数与总数相等 | | |
| 设计约束 | -无- | | |
| 数据校验 | -无- | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

**访问统计**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | FUN- 6.2 | 功能名称 | 访问统计 |
| 功能描述 | 流量统计（统计PV 和 UV）、来源统计（统计来源URL）、受访分析（页面的访问量排行）、地域分析（访问的地理区域）、浏览器统计、操作系统分析、终端统计（访问网站使用的设备）、详细访问日志、站点访问统计（按站点统计PV和 UV）、栏目访问统计（按栏目统计 PV 和 UV） | | |
| 数据描述 | 【访问记录表】（whir\_dev\_visitlog）。 | | |
| 业务约束 | 前台页面模板需要调用统计脚本代码。 | | |
| 设计约束 | -无- | | |
| 数据校验 | -无- | | |
| 异常处理 | 创建失败，不写入数据 | | |
| 权限描述 | -无- | | |
| 日志描述 | -无- | | |
| 消息提示 | -无- | | |

### 基金超市适配

#### 内容架构适配

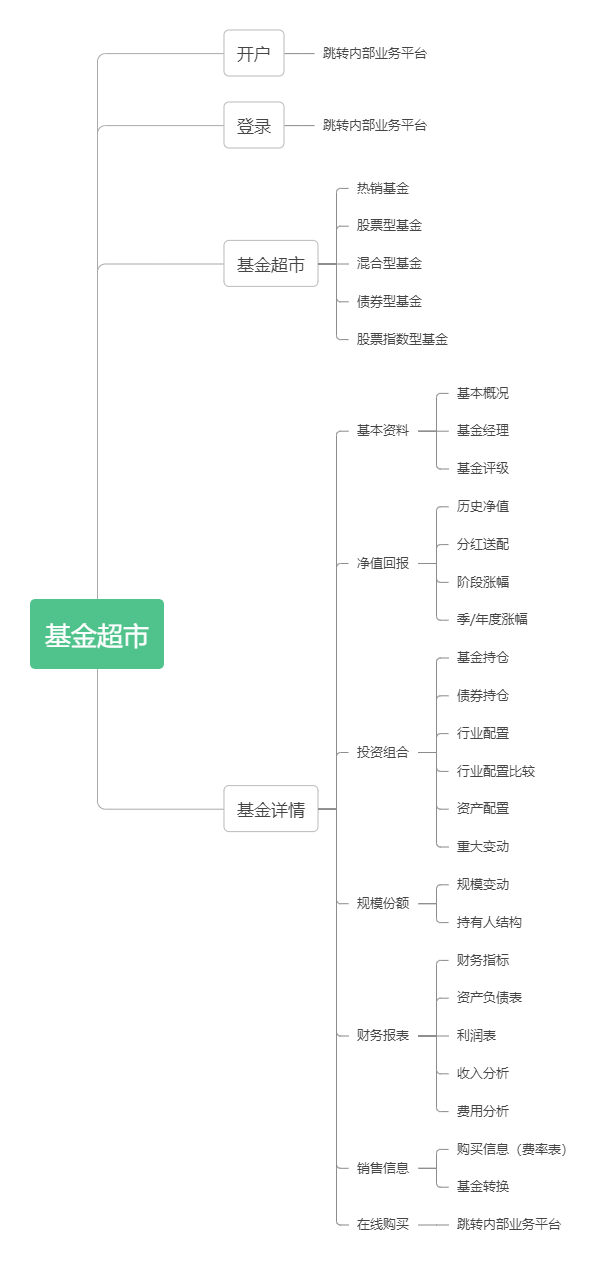
保留现有基金超市的内容架构，并根据双方具体沟通并确认的最终内容架构为准建设基金超市系统。

#### 前端页面适配

根据现有基金超市系统数据呈现逻辑并结合行业的流行趋势，适配基金超市前台页面，按照数据呈现逻辑规划页面设计及内容指引，提升基金超市的整体访问体验。

#### 功能适配

具体所需适配迁移功能如下，注册、登录及购买操作，通过跳转内部业务系统实现，其他功能可选择4.5.2.4方式实现数据的呈现。



#### 数据接口适配

贵公司提供数据获取接口及数据爬取软件接口（或第三方平台接口），我司对数据接口进行适配，将数据爬取软件获取的相关数据存入数据库并设计数据呈现页面。

### 适老化建设及适配

#### 建设依据

《残疾人权利公约》

《中国残疾人事业“十二五”发展纲要》

工业和信息化部及中国残疾人联合会联合发布了工信部联管函〔2020〕146号《关于推进信息无障碍的指导意见》

国务院办公厅印发《关于切实解决老年人运用智能技术困难的实施方案》的通知

#### 功能详情

信息适老化：适合老年人。一般指在互联网（网站、移动端）建设或设计时进行适老化，满足进入老年生活的人群获取、利用和使用信息的需求。

放大（大字体、大图标、大按钮、大鼠标）；

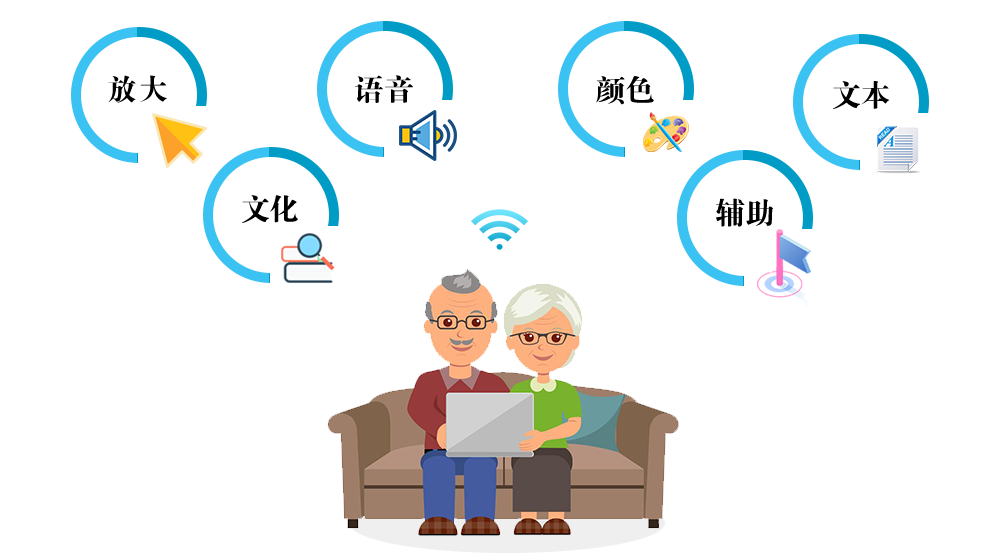
语音（语音提示、语音朗读、语音控制）；

颜色（白底黑字、蓝底黄字、黄底黑字、黑底黄字）；

文本（纯文本、去装饰广告）；

文化（简体、繁体、拼音、少数民族语）；

辅助（十字定位、畏光灰化）；



**无障碍辅助工具**

是主要为部分视力障碍人群（不含盲人）提供的用来辅助浏览的模块。该功能为老年人提供字体放大缩小、页面放大缩小功能。为斜视人群提供辅助线功能。为色盲色弱人群提供色彩高对比度功能。同时还提供放大镜、简繁转换、拼音标注、语音朗读等辅助获取信息的功能

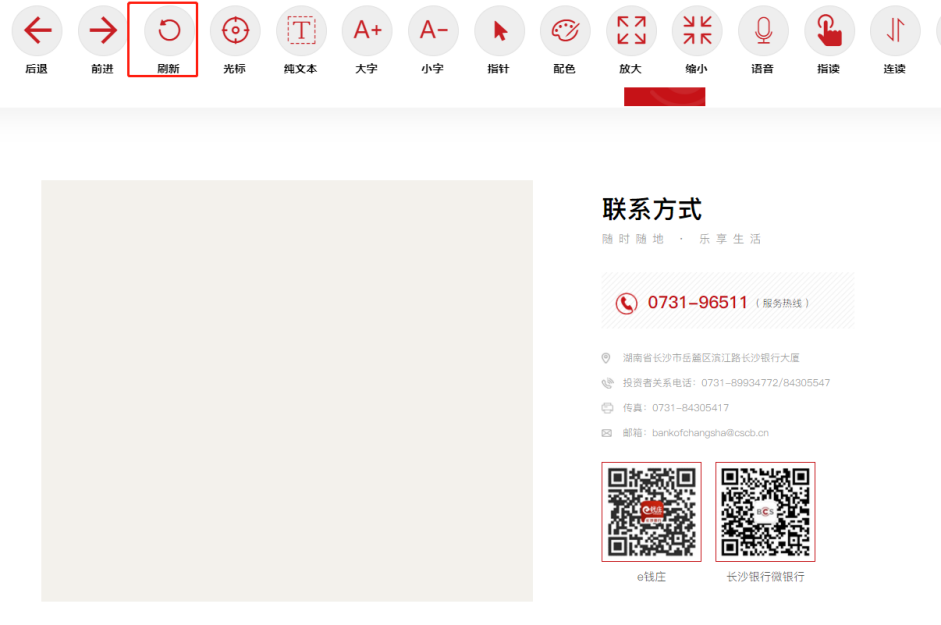


**功能详情**

首页：点击首页可返回网站首页



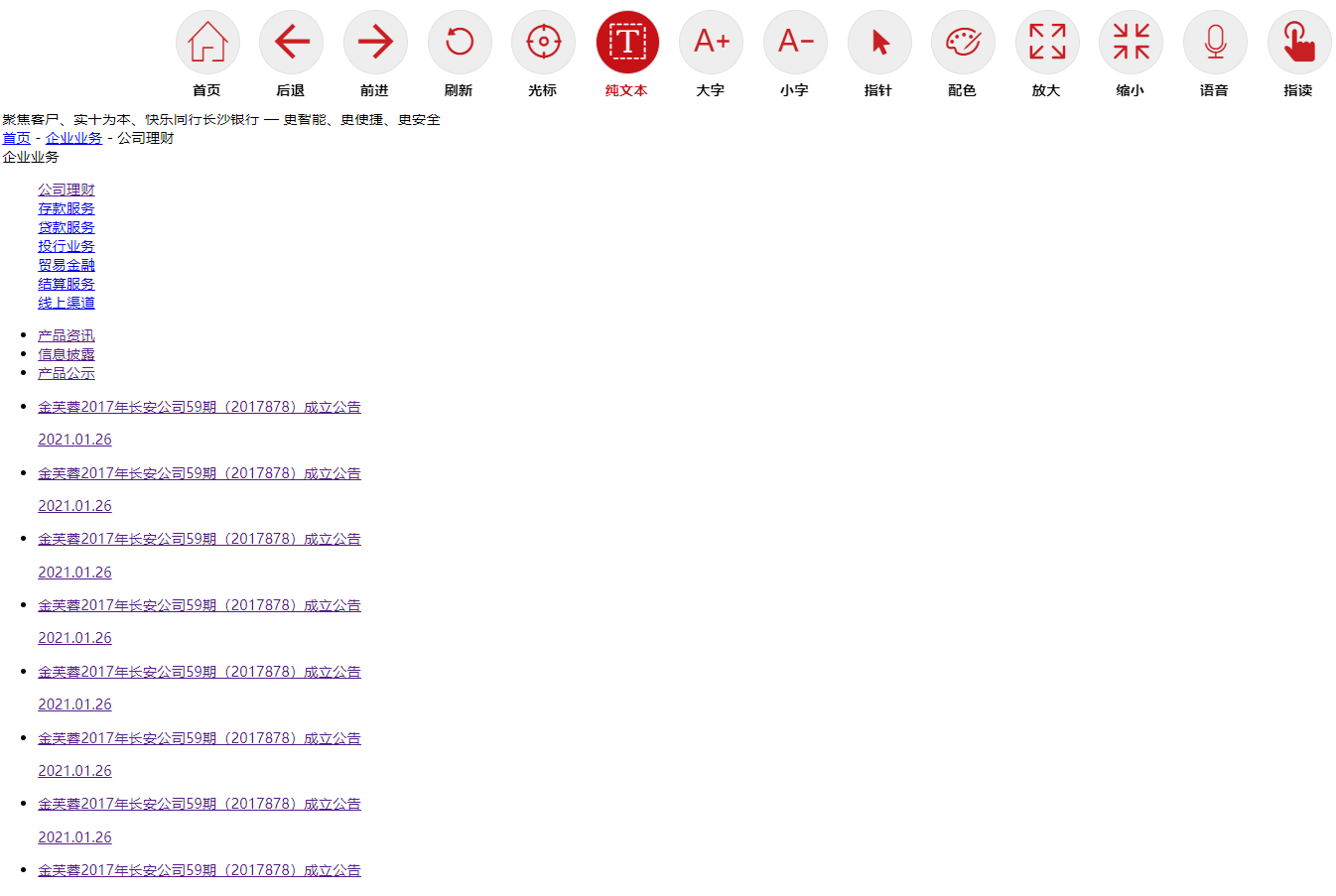
刷新：点击刷新当前页面



后退/前进：点击后退或前进页面



纯文本：页面进入纯文本状态，该状态下页面的所有样式都会取消



箭头：改变当前鼠标样式，方便老年人、视障碍人士使用。



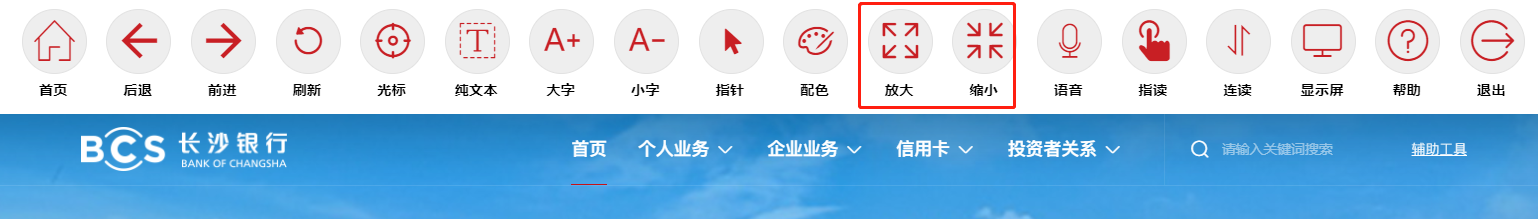
光标：是横竖两条红色的基准线，为视力障碍用户校队阅读的位置提供了帮助。



配色：为色彩浏览障碍者提供不同的配色的页面浏览方式。



页面：页面放大缩小功能，将网站当前页面显示比例放大、缩小。





语音：语音功能可进行语音打开/关闭设置；进行指读、连读设置；进行语速加快/减慢设置。







开启指读后，系统会自动读取鼠标指针当前位置内容，当指针放置在无障碍辅助工具的相关按钮时，系统会朗读该功能按钮名称，辅助用户使用其他功能



当指针放置在网站内容区域时，系统会朗读网站该功能名称，辅助用户使用网站相关功能，如当鼠标放置在检索框时，系统会朗读该功能为：检索框-请输入关键词搜索



当指针放置在网站内容区域时，系统会朗读该区域内容，辅助用户获取网站信息，如下图，当指针放置具体内容时，该内容会变色，系统会朗读变色区域内容



开启连读功能后，系统会自动从当前页面首个内容进行连续阅读适用于新闻资讯类详情页面，便于用户通过声音获取网站的详细内容



字体大小：通过点击字体+、字体小控制页面字体大小。



显示屏：在页面下方出现一个现实区域，该区域会显示当前鼠标所指到的文字信息。显示屏可以进行简体字转繁体字的显示，也可以打开拼音显示。



## 适配验证

### 适配验证规划

在完成应用改造与适配工作后，各应用与产品厂商及承建单位需针对测试结果进行分析、优化，问题修复后按照适配测试流程进行回归测试，直到所有问题均得到解决并经承建单位确认后，应用改造与适配工作方可结束。适配验证内容及通过准则如下：

适配验证表

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **测试内容** | **测试方法** | **通过准则** |
| 1 | 操作系统安装测试 | 在服务器及桌面终端上通过光盘安装指定版本操作系统 | 1、操作系统安装过程无错误、中断等异常情况发生；  2、安装后的操作系统可使用root用户、普通用户登录； |
| 2 | 基础软件安装测试 | 在操作系统上安装指定基础软件，记录安装结果 | 1、安装过程无错误；  2、root用户、普通用户可正常使用软件；  3、软件可正常卸载； |
| 3 | 基础软件功能测试 | 在指定操作系统平台上使用root用户、普通用户操作基础软件 | 1、单机环境功能可正常使用；  2、集成环境下功能可正常使用； |
| 4 | 外设功能测试 | 在操作系统平台上安装外设驱动，用root用户、普通用户操作外设 | 1、单机环境功能可正常使用；  2、集成环境下功能可正常使用； |
| 5 | 应用系统功能测试 | 在终端使用指定浏览器访问应用系统，并对基础功能进行验证 | 1、应用系统可正常访问；  2、基础功能运行正常； |
| 6 | 应用系统性能测试 | 通过X86终端向服务器发送并发请求，验证服务器处理并发请求能力； | 1、支持多用户并发操作  2、多用户并发情况下系统运行稳定，系统资源不出现瓶颈； |
| 7 | 办公套件性能测试 | 在单机环境下运行办公套件，测试指定大小文件的读取时间 | 1、读取时间在可接受范围内；  2、大数据量操作情况下软件运行稳定； |

### 国产计算机整机验证

对采用国产处理器的自主计算机进行与国产固件、操作系统、外设等进行适配验证，形成可用的自主计算机整机设备清单和可适配的外设型号清单。主要验证内容包括计算机单机基本配置验证：主要接口标准符合性和稳定性验证,与国产固件之间的适配性验证；与国产操作系统之间的适配性验证；与常用的板卡和外设的适配验证。

### 国产基础软件适配

国产基础软件适配优化针对办公应用场景，开展桌面操作系统、流版签、浏览器、底层基础软件及常用外设自主可控终端优化适配，解决自主可控终端性能低，稳定性差，难以满足日常办公应用需求的问题。从操作系统开关机时间、典型流板式文件打开时间、典型网页加载时间、混合文件拷贝时间等方面提升终端平台的性能和易用性。

开展桌面操作系统、流版签、浏览器、底层基础软件等国产软件及国产打印机、扫描仪等外设在国产化终端平台上的适配优化工作，突破基础库硬件加速、内存驻留、软件热点分析、软件架构优化、软件性能瓶颈分析、内核调优等关键技术，提升操作系统及基础办公软件运行效率，通过迭代优化实现终端平台的整体性能逐步提升。

### 应用软件适配验证

#### 整体性验证

1．系统功能

各系统此次升级将继续覆盖原有系统功能的建设，同时保持原有功能的使用逻辑，并且针对每个功能模块的功能名称、功能作用设定不变，延续工作人员对各系统各个操作环节的熟悉度，使工作人员可以快速的适应新搭建的系统。

2．管理用户同步

国产化各系统的管理人员信息（包括管理人员详情、组织、角色、权限等）和原有各系统中管理人员信息保持一致，同时针对每个管理人员的登录账号保持不变。

#### 兼容性验证

由于系统部署运行环境从CPU、操作系统、数据库、中间件、办公套件等一系列环境的改造，与之相关的应用功能在读取数据、使用组件等都会发生变化，系统功能兼容性验证显得尤为重要。

#### 性能验证

在国产化环境下，系统各功能模块的实际性能表现，是给与用户最为直观的感受体验，系统的卡顿会导致客户是去使用系统的耐心，这也违背了国产化升级和信息化建设的必要性。在适配迁移后，需要进行压力测试，对用户登录、表单打开、流程发送等诸多环节进行全面的性能测试及验证，针对性能验证结果，给与必要的硬件配置提升及性能调优工作。

#### 业务流畅性验证

国产化环境适配后，部分依赖非国产化环境的功能会无法使用，会导致客户在使用系统时，对某些功能环节的流畅性提出质疑。在适配迁移工作后，需要对业务流程进行全流程测试验证工作，用以发现业务流程具体环节中的功能缺失，按照国产化环境现状，积极为用户建立可行的替代功能。

#### 数据迁移验证

在系统数据迁移工作进行后，必须对数据迁移进行工作验证，及时梳理遗漏丢失的数据，利用人工相结合的方式，将遗漏的数据的进行补充完善，确保数据的一致性。

#### 接口可用性验证

由于系统需要进行安全可靠环境下的开发和迁移改造，系统与其他各业务系统间的各类接口就需要重新进行适配调整，继续保证接口的有效性。

## 优化策略

### 部署优化

项目建设初期，用户访问量有限，数据量不大，多数系统采用单台服务器部署应用服务，系统服务、文件、数据库等所有系统资源部署在一台服务器上。

应用和数据分离

随着用户量和数据量的不断攀升，业务对系统的性能要求越来越高，这是需要将应用和数据分离，单独部署相关的业务组件。

### 缓存优化

系统的瓶颈还是在数据库方面，需要减少数据库的查询次数。 就利用缓存。

查询出来某类数据放在Map里，下次需要的时候，直接从Map里取，取不到的时候再查询数据库。缓存分为两种全局缓存，局部缓存。

#### 全局缓存

此类情况，适用于不经常变化的数据，例如基础设置信息，流程设置信息等等。可以采用单例（Singleton）模式，保证一个类仅有一个实例，并提供一个访问它的全局访问点。取出来的数据放在这个单例类中，下次需要数据数，直接从这个单例类中取，取不到数据时，再从数据库中取，并且放置在单例类中。全局缓存有一个问题就是如果设置信息有了变化，从单例类中取到数据就是老的，不准确。解决方案是：修改基础设置信息时，同时更改单例类中的数据。

#### 局部缓存

此类情况是，某次复杂操作，在多个地方需要多次查询一张表中的数据。每次可能是查询条件不同或者其它原因。

可以先按照某个字段在数据库中查询出比较大范围的数据集。然后在需要查询数据的地方，在这个数据集里过滤出需要的数据，避免多次查询数据库。

### 设计优化

#### 合并与压缩前台js,css等静态文件

如果某个页面包含的js文件，css文件，比较多，比较大，可以考虑把js,css合并为一个文件。并且压缩。

比如工作流设计器页面，包含几十个js文件。经过合并与压缩js文件后，前台打开页面有很大改善。

可以使用相关工具来合并与压缩js文件。此工具不带混淆功能，只合并与压缩功能。

#### 异步处理

比如打开的某个页面，有基本信息，日志信息等等，先取出基本信息，点击日志信息时，再到后台去取数据，不要一次查询出来

#### 动态页面静态化

某些页面一般不怎么变化，可以考虑静态化。减少编译与执行时间。比如工作流某个包含页，里面是js 国际化内容。国产化环境不需要国际化。可以直接改为静态页面。减少编译与执行时间。

### 数据库优化

#### 加索引优化SQL查询速度

通过debug日志算出单个sql操作或者方法的时间，或者每个方法开始结束处记录时间，方法结束时，debug或者打印出耗时。也可以通过数据库系统日志查找出比较出耗时的sql，这个是总个系统耗时比较长的sql,一般在这里查找某个功能的耗时比较难。

#### 换查询条件优化SQL查询速度

国产化数据库，由于性能方面局限，有的表按A字段查询加了索引还是很慢，但是按B字段查加了索引比较快。就更改sql 按B字段查询出结果，然后用程序过滤满足A条件的结果集。

#### 不利用数据库生成主键

在并发性大的情况下，利用数据库的序列生成主键，比较慢，插入数据条数比较多时，光取主键就花了很长时间。

更改主键生成方式：例如利用java.util.UUID.randomUUID().toString()等方式生成主键,优化插入数据的速度。

#### 采用PreparedStatement方式

采用PreparedStatement不仅可以防止sql注入，还可减少sql编译时间，降低系统消耗。最新的Page 、DbUtil 等公用类都可以使用PreparedStatement方式，请按规范使用。其它在ejb里使用的SQL,HQL都尽可能改为PreparedStatement 方式

#### 优化代码，减少数据库查询次数

查找系统中原来代码中，不优化的代码，减少数据库查询次数。

## 系统测试

### 准备工作

#### 系统基础功能验证

性能测试在什么阶段适合实施？切入点很重要！一般而言，只有在系统基础功能测试验证完成、系统趋于稳定的情况下，才会进行性能测试，否则性能测试是无意义的。

#### 测试团队组建

根据该项目的具体情况，组建一个几人的性能测试team，其中DBA是必不可少的，然后需要一至几名系统开发人员（对应前端、后台等），还有性能测试设计和分析人员、脚本开发和执行人员；在正式开始工作之前，应该对脚本开发和执行人员进行一些培训，或者应该由具有相关经验的人员担任。

#### 工具的选择

综合系统设计、工具成本、测试团队的技能来考虑，选择合适的测试工具，最起码应该满足一下几点：

①支持对web（这里以web系统为例）系统的性能测试，支持http和https协议；

②支持对webserver、前端、数据库的性能计数器进行监控；

#### 预先的业务场景分析

为了对系统性能建立直观上的认识和分析，应对系统较重要和常用的业务场景模块进行分析，针对性的进行分析，以对接下来的测试计划设计进行准备。

### 测试计划

测试计划阶段最重要的是分析用户场景，确定系统性能目标。

#### 性能测试领域分析

根据对项目背景，业务的了解，确定本次性能测试要解决的问题点；是测试系统能否满足实际运行时的需要，还是目前的系统在哪些方面制约系统性能的表现，或者哪些系统因素导致系统无法跟上业务发展。确定测试领域，然后具体问题具体分析。

#### 确定性能目标

前面已经确定了本次性能测试的应用领域，接下来就是针对具体的领域关注点，确定性能目标（指标）；其中需要和其他业务部门进行沟通协商，以及结合当前系统的响应时间等数据，确定

最终我们需要达到的响应时间和系统资源使用率等目标；比如：

①登录请求到登录成功的页面响应时间不能超过2秒；

②报表审核提交的页面响应时间不能超过5秒；

③文件的上传、下载页面响应时间不超过8秒；

④服务器的CPU平均使用率小于70%，内存使用率小于75%；

⑤各个业务系统的响应时间和服务器资源使用情况在不同测试环境下，各指标随负载变化的情况等；

#### 制定测试计划的实施时间

预设本次性能测试各子模块的起止时间，产出，参与人员等等。

### 测试脚本设计与开发

性能测试中，测试脚本设计与开发占据了很大的时间比重。

#### 测试环境设计

本次性能测试的目标是需要验证系统在实际运行环境中的性能外，还需要考虑到不同的硬件配置是否会是制约系统性能的重要因素。因此在测试环境中，需要部署多个不同的测试环境，在不同的硬件配置上检查应用系统的性能，并对不同配置下系统的测试结果进行分析，得出最优结果（最适合当前系统的配置）。

这里所说的配置大概是如下几类：

①数据库服务器

②应用服务器

③负载模拟器

④软件运行环境，平台

测试环境测试数据，可以根据系统的运行预期来确定，比如需要测试的业务场景，数据多久执行一次备份转移，该业务场景涉及哪些表，每次操作数据怎样写入，写入几条，需要多少的

测试数据来使得测试环境的数据保持一致性等等。

可以在首次测试数据生成时，将其导出到本地保存，在每次测试开始前导入数据，保持一致性。

#### 测试场景设计

通过和业务部门沟通以及以往用户操作习惯，确定用户操作习惯模式，以及不同的场景用户数量，操作次数，确定测试指标，以及性能监控等。

#### 测试用例设计

确认测试场景后，在系统已有的操作描述上，进一步完善为可映射为脚本的测试用例描述，用例大概内容如下：

用例编号：查询表单\_xxx\_x1（命名以业务操作场景为主，简洁易懂即可）

用例条件：用户已登录、具有对应权限等；

操作步骤：

①进入对应页面，

②查询相关数据，

③勾选导出数据，

④修改上传数据。

#### 脚本和辅助工具的开发及使用

按照用例描述，可利用工具进行录制，然后在录制的脚本中进行修改；比如参数化、关联、检查点等等，最后的结果使得测试脚本可用，能达到测试要求即可；

### 测试执行与管理

在这个阶段，只需要按照之前已经设计好的业务场景、环境和测试用例脚本，部署环境，执行测试并记录结果即可。

#### 建立测试环境

按照之前已经设计好的测试环境，部署对应的环境，由运维或开发人员进行部署，检查，并仔细调整，同时保持测试环境的干净和稳定，不受外来因素影响。

#### 执行测试脚本

这一点比较简单，在已部署好的测试环境中，按照业务场景和编号，按顺序执行我们已经设计好的测试脚本。

#### 测试结果记录

根据测试采用的工具不同，结果的记录也有不同的形式；现在大多的性能测试工具都提供比较完整的界面图形化的测试结果，当然，对于服务器的资源使用等情况，可以利用一些计数器或

第三方监控工具来对其进行记录，执行完测试后，对结果进行整理分析。

### 测试分析

#### 测试环境的系统性能分析

根据我们之前记录得到的测试结果（图表、曲线等），经过计算，与预定的性能指标进行对比，确定是否达到了我们需要的结果；如未达到，查看具体的瓶颈点，然后根据瓶颈点的具体数据，

进行具体情况具体分析（影响性能的因素很多，这一点，可以根据经验和数据表现来判断分析）。

#### 硬件设备对系统性能表现的影响分析

由于之前设计了几个不同的测试环境，故可以根据不同测试环境的硬件资源使用状况图进行分析，确定瓶颈是再数据库服务器、应用服务器抑或其他方面，然后针对性的进行优化等操作。

#### 其他影响因素分析

影响系统性能的因素很多，可以从用户能感受到的场景分析，哪里比较慢，哪里速度尚可。

至于其他诸如网络带宽、操作动作、存储池、线程实现、服务器处理机制等一系列的影响因素，具体问题具体分析。

#### 测试中发现的问题

在性能测试执行过程中，可能会发现某些功能上的不足或存在的缺陷，以及需要优化的地方，这也是执行多次测试的优点。

# 信创技术与软硬件配置建议

## 信创hcs主打材料

|  |  |  |
| --- | --- | --- |
| **序号** | **产品类别** | **产品名称** |
| **1** | CPU | 飞腾 |
| **2** | 兆芯 |
| **3** | 海光 |
| **4** | 鲲鹏 |
| **5** | 申威 |
| **6** | 龙芯 |
| **7** | 操作系统 | 银河麒麟 |
| **8** | 中科方德 |
| **9** | 欧拉 |
| **10** | 深度 |
| **11** | 中标麒麟 |
| **12** | 麒麟 |
| **13** | UOS |
| **14** | 普华 |
| **15** | 数据库 | 人大金仓 |
| **16** | 达梦数据库 |
| **17** | 神通数据库 |
| **19** | 海量 |
| **22** | mogdb |
| **25** | 南大通用 |
| **27** | 中间件 | 东方通 |
| **28** | 金蝶 |
| **29** | 宝兰德 |
| **30** | 中创 |
| **31** | 普元 |
| **32** | 华宇 |
| **33** | 流式编辑 | 金山WPS专业版客户端 |
| **34** | 金山WPS在线编辑服务 |
| **35** | 永中office |
| **36** | 永中在线编辑服务 |
| **37** | 流式预览 | 金山WPS在线预览服务 |
| **38** | 金山WPS专业版客户端 |
| **49** | 永中DCS/FCS服务 |
| **40** | 永中office |
| **41** | 福昕轻阅读服务V2.0（需二开） |
| **42** | 数科轻阅读服务V3.0（需二开） |
| **43** | 版式预览 | 金山WPS在线预览服务 |
| **44** | 永中DCS/FCS服务 |
| **45** | 福昕轻阅读服务V2.0 |
| **46** | 福昕客户端产品V8.0 |
| **47** | 数科轻阅读服务V3.0 |
| **48** | 数科客户端产品V3.0 |
| **49** | 格式转换 | 金山WPS在线预览服务 |
| **50** | 永中DCS/FCS服务（需要单独采购OFD授权） |
| **51** | 福昕OFD转换V1.0 |
| **52** | 数科OFD转换V3.0 |

## 软硬件资源配置建议

### 服务器配置建议清单

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **服务器** | **数量** | **操作系统** | **CPU参考** | **CPU个数** | **内存** | **本地磁盘** | **备注** |
| 网站应用服务器 | 2台 | / | / | 1颗32核 | 32G | 500G SSD | 统一部署 |

### 软件配置建议清单

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **类型** | **数量** | **备注说明** |
| **1** | 数据库 | 2套 | 外网门户数据库2套 |
| **2** | 中间件 | 1套 | 外网门户1套 |
| **3** | 操作系统 | 4套 | 外网门户4套 |
| **4** | 浏览器 | / | / |
| **5** | 流式软件 |  |  |
| **6** | 版式软件 |  | 用来查看下载后的OFD文件 |

### 负载均衡建议清单（软负载/可选）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **操作系统** | **CPU参考** | **CPU个数** | **内存** | **本地磁盘** |
| / | / | 1颗32核 | 32G | 300G（SSD） |

### 服务器操作系统分区建议

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **服务器类型** | **服务器路径** | **分区大小** | **分区格式** | **备注** |
| 各应用  服务器 | 交换区 | 16g | swap |  |
| / | 64g | ext4 |  |
| 程序文件所在的路径（/opt） | 剩余空间 | ext4 | 根据实际程序存放路径调整 |
| 各数据库  服务器 | 交换区 | 32g | swap |  |
| / | 64g | ext4 |  |
| 数据库程序和数据文件所在路径(/opt) | 剩余空间 | ext4 | 根据实际程序存放路径调整 |