分布式机群管理系统实验报告

**小组成员:**

SA20225085 朱志儒

SA20225084 朱康乐

SA20225087 邹绍强

目录

[1 可行性分析 1](#_Toc61002983)

[1.1 技术可行性分析 1](#_Toc61002984)

[1.1.1 前端技术 1](#_Toc61002985)

[1.1.2 后端技术 1](#_Toc61002986)

[1.2 经济可行性分析 1](#_Toc61002987)

[1.3 时间可行性分析 1](#_Toc61002988)

[2 实验简介 2](#_Toc61002989)

[2.1 实验目的 2](#_Toc61002990)

[2.2 实验内容 2](#_Toc61002991)

[2.3 实验结构图 2](#_Toc61002992)

[3 需求分析 3](#_Toc61002993)

[3.1 导出需求 3](#_Toc61002994)

[3.2 分析建模 4](#_Toc61002995)

[3.2.1 用例视图 4](#_Toc61002996)

[3.2.1.1 用例图 4](#_Toc61002997)

[3.2.1.2 查看说明用例描述 4](#_Toc61002998)

[3.2.1.3 配置管理用例描述 5](#_Toc61002999)

[3.2.1.4 浏览节点信息用例描述 5](#_Toc61003000)

[3.2.1.5 节点实时状态浏览用例描述 5](#_Toc61003001)

[3.2.1.6 节点历史状态浏览用例描述 6](#_Toc61003002)

[3.2.2 逻辑视图 6](#_Toc61003003)

[3.2.2.1 静态-类图 6](#_Toc61003004)

[3.2.2.2 动态-顺序图 7](#_Toc61003005)

[4 设计 10](#_Toc61003006)

[4.1 视图 10](#_Toc61003007)

[4.1.1 开发视图 10](#_Toc61003008)

[4.1.2 进程视图 11](#_Toc61003009)

[4.1.3 部署视图 12](#_Toc61003010)

[4.2 系统架构概况 13](#_Toc61003011)

[5 实现 15](#_Toc61003012)

[5.1 代码结构 15](#_Toc61003013)

[5.1.1 系统总代码结构 15](#_Toc61003014)

[5.1.2 机器节点端代码 15](#_Toc61003015)

[5.1.3 服务器端代码 16](#_Toc61003016)

[5.2 运行结果 16](#_Toc61003017)

[6 测试结果 17](#_Toc61003018)

[6.1 功能性测试结果 17](#_Toc61003019)

[6.1.1 单元测试结果 17](#_Toc61003020)

[6.1.2 集成测试结果 17](#_Toc61003021)

[6.1.3 确认测试 17](#_Toc61003022)

[6.2 非功能性测试结果 17](#_Toc61003023)

[6.2.1 性能测试结果 17](#_Toc61003024)

[6.2.2 可扩展性测试结果 17](#_Toc61003025)

[6.2.3 灵活性测试结果 18](#_Toc61003026)

# 可行性分析

## 技术可行性分析

### 前端技术

现有html,css,javaScript技术以及VUE框架. 所以在用户浏览器上显示界面可行.

### 后端技术

现有java语言,servlet,以及springboot框架,tomcat服务器. 所以搭建一个含有动态部分的网站是可行的. 而且通过http协议, 服务器采用了netty框架满足多用户并发访问需求也能实现机器节点向服务器的信息发送,所以技术上是可行的.

## 经济可行性分析

实验几乎没有开销,所以经济上是可行的.

## 时间可行性分析

实验时间为16周, 时间上可行.

# 实验简介

## 实验目的

以一个分布式机群监管系统作为例子，要求学生掌握使用各种结构视图，实现对系统结构的描述，并采用各种开源框架和具体编程技术实现该系统的简化版本。

## 实验内容

根据用户需求列表，整理用户需求，完成用例图分析。根据附录提供的结构图，采用课程所描述的各种视图方法，完成对系统结构的描述。最后编程实现系统。

## 实验结构图



图1 结构图

# 需求分析

在此部分主要考虑系统的功能性, 设计出了软件体系结构中4+1视图中的用例视图与逻辑视图.

## 导出需求

功能性需求,如表1所示

表格1 用户需求列表

|  |  |  |
| --- | --- | --- |
| **序号** | **需求** | **内容** |
| 一 | 服务器功能模块 | 多用户的B/S架构，数据库暂时采用MySql |
| 1. | 控制命令传输 | 基于标准Http协议，实现控制命令发送 |
| 2. | 机器状态接收 | 基于标准Http协议，实现门禁状态信息的接收 |
| 3. | 管理配置 | 实现IP地址、同步信息的更新频率等各种设置信息的远程配置 |
| 4. | 数据查询 | 实现各种状态数据的查询汇总等 |
| 5. | 报表 | 实现以EXCEL、PDF等格式保存打印报表等 |
| 6. | 浏览器信息推送 | 有状态变化时向浏览器发送状态改变的信息 |
| 7. | 浏览器界面 | 采用JavaScript实现富客户端界面，如站点树、圆饼图、菜单等 |
| 8. | 心跳 | 接收基站控制器心跳数据信息，确保其是否在线 |
|  |  |  |
| 二 | 分布式机器 |  |
| 1. | 控制命令 | 基于标准协议接收服务器发送的指令，实现基本控制 |
| 2. | 状态上传 | 基于标准协议实现各种机器状态的上传 |
| 3. | 远程配置 | 实现远端对IP地址、以及各种管理数据的配置 |
| 4. | 数据传输 | 因为本方案采用的是WEB服务器为中心的B/S架构，因此拟采用Rest和Web Socket等协议实现和服务器间的数据传输，具有非常好的可扩展性和双向交互能力。 |
| 5. | 心跳 | 实现对服务器的心跳信息控制 |

根据需求列表整理得到，管理员可以通过网站进行配置管理，查看说明，浏览节点信息。其中浏览节点信息功能可以分为三个子用例，节点实时状态浏览、实时状态显示、历史状态图。节点的心跳信息打包通过HTTP协议发送到服务器上，服务器解析心跳包，获得实时状态信息并保存。前端采用VUE框架实现富客户端界面，输出节点内存占有率的折线图。

非功能性需求,如表2所示.

|  |  |
| --- | --- |
| 性能需求: | 响应时间迅速, 数据传输快速 |
| 可扩性需求: | 链接的站点能够扩展到几千个，能满足多用户并发访问 |
| 灵活性需求： | 提供外部访问接口，能够实现后期如手机应用、外部应用的调用。程序升级简单 |

## 分析建模

### 用例视图

#### 用例图



图 用例图

#### 查看说明用例描述

|  |  |
| --- | --- |
| 用例名称 | 查看说明 |
| 概述 | 查看分布式机群监管系统的概述 |
| 依赖 | 无 |
| 参与者 | 机房管理员 |
| 前置条件 | 无 |
| 主序列描述 | 1. 用户访问机房监管系统的网站 2. 用户点击 查看说明 |
| 可替换序列描述 | 无 |
| 后置条件 | 网站显示概述 |

#### 配置管理用例描述

|  |  |
| --- | --- |
| 用例名称 | 配置管理 |
| 概述 | 配置机器节点 |
| 依赖 | 无 |
| 参与者 | 机房管理员 |
| 前置条件 | 无 |
| 主序列描述 | 1. 用户访问机房监管系统的网站 2. 用户点击配置管理 3. 用户输入机器节点信息 4. 用户点击确认 |
| 可替换序列描述 | 无 |
| 后置条件 | 修改机器配置成功 |

#### 浏览节点信息用例描述

|  |  |
| --- | --- |
| 用例名称 | 浏览节点信息 |
| 概述 | 可以浏览节点的实时信息,历史信息 |
| 依赖 | 包含节点实时状态浏览, 节点历史状态浏览 |
| 参与者 | 机房管理员 |
| 前置条件 | 无 |
| 主序列描述 | 1. 用户访问机房监管系统的网站 2. 用户点击浏览节点信息 3. 用户可选择查看实时信息或者历史信息 |
| 可替换序列描述 | 无 |
| 后置条件 | 显示查看实时状态与历史信息 |

#### 节点实时状态浏览用例描述

|  |  |
| --- | --- |
| 用例名称 | 实时状态浏览 |
| 概述 | 查看当前已经连接系统的机器节点实时状态 |
| 依赖 | 被 浏览节点信息用例 包含 |
| 参与者 | 机房管理员 |
| 前置条件 | 1. 用户访问机房监管系统的网站 2. 用户点击浏览节点信息 |
| 主序列描述 | 1. 用户点击浏览节点实时状态 2. 系统在网页上显示节点实时状态,如:cpu占用率等 |
| 可替换序列描述 | 无 |
| 后置条件 | 显示了节点的实时状态图。 |

#### 节点历史状态浏览用例描述

|  |  |
| --- | --- |
| 用例名称 | 历史状态浏览 |
| 概述 | 查看当前已经连接系统的机器节点历史状态 |
| 依赖 | 被 浏览节点信息用例 包含 |
| 参与者 | 机房管理员 |
| 前置条件 | 1. 用户访问机房监管系统的网站 2. 用户点击浏览节点信息 |
| 主序列描述 | 1. 用户点击浏览节点历史状态 2. 系统在网页上显示节点历史状态,如:cpu占用率等 |
| 可替换序列描述 | 无 |
| 后置条件 | 显示了节点的历史状态图。 |

### 逻辑视图

#### 静态-类图

根据导出需求完成的需求列表,找出需求分析中所有的类,并建立类之间联系.如图2所示.



图2 类图

#### 动态-顺序图

根据静态建模得出的类图以及系统的用例, 绘制了不同用例的顺序图.

配置管理用例的顺序图, 如图3所示.

查看节点实时状态用例的顺序图, 如图4所示.

查看节点历史状态用例的顺序图, 如图5所示.



图3 配置管理顺序图



图4 浏览实时状态顺序图



图5 浏览历史状态顺序图

# 设计

在此部分主要考虑系统的非功能性即质量属性部分, 设计出了软件体系结构中4+1视图中的开发视图, 进程视图和部署试图.

## 视图

### 开发视图

根据用例要求的功能,对逻辑视图中的类进行分类,并组成包的形式.用于划分模块. 具体的包以及模块如图6所示.



图6 开发视图

### 进程视图

分析考虑实际系统运行中所有的机器上运行的关于本系统的进程. 设计图7所示的进程视图.



图7 进程视图

### 部署视图

分析考虑实际系统运行中所有的机器以及其之间的联系和部署. 设计图8所示的部署视图.



图8 部署视图

## 系统架构概况

根据实验要求,以及需求和设计部分的工作,本次实验对系统结构的描述如图9所示.



图9 系统结构图

# 实现

## 代码结构

具体代码文件见代码压缩包.

### 系统总代码结构

系统代码结构包含

静态页面部分:dcss-ui;

服务器端:sever

机器节点端:client

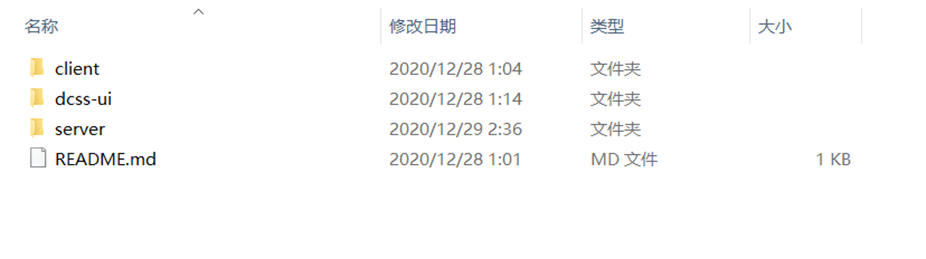
如图10所示

图10 系统代码结构

### 机器节点端代码

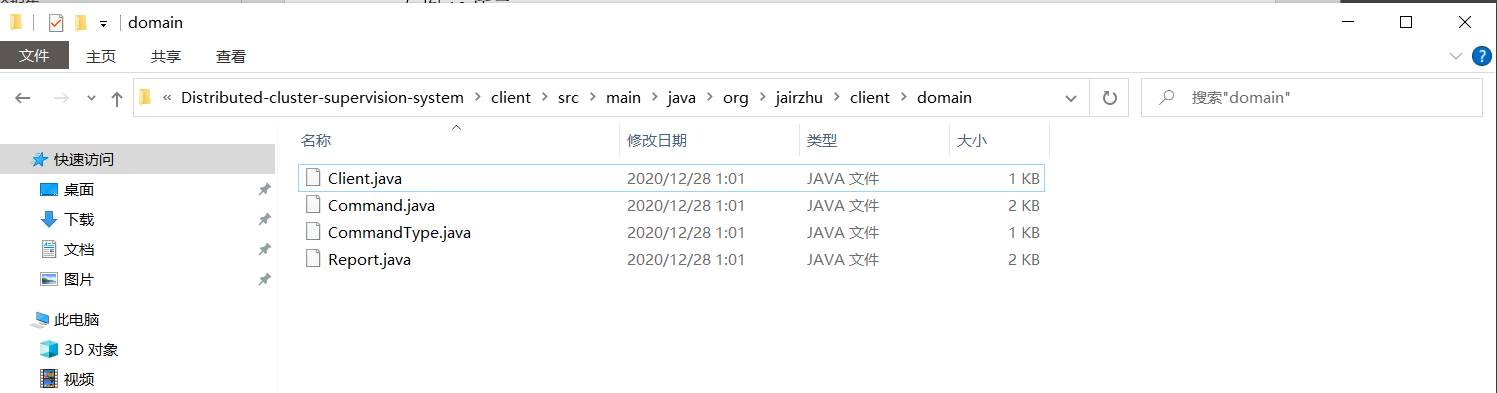
机器节点端即client端,包含client类,command类,和报告类. 如图11 所示

图11 机器节点的模拟实现

### 服务器端代码

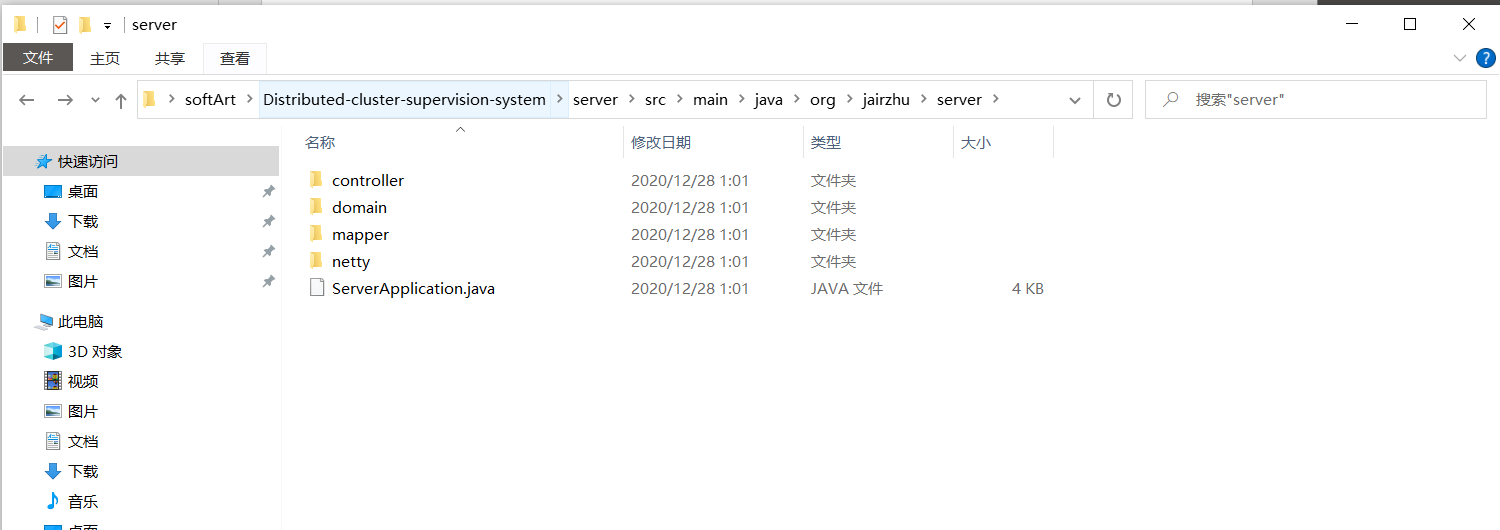
服务器端包含controller, domain, mapper, netty. 具体结构如图12 所示.

图12 服务器端部分代码结构

## 运行结果

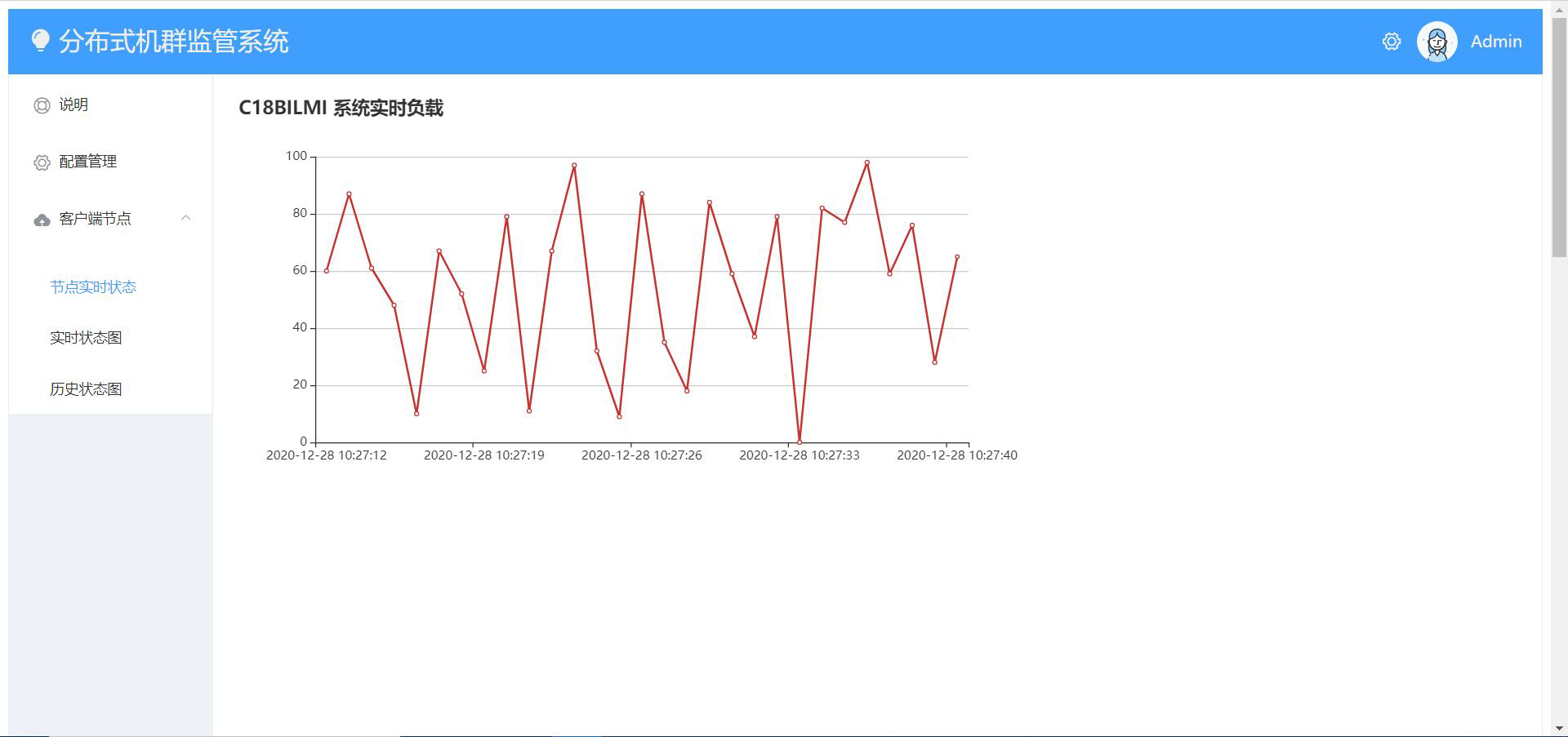
运行时的截图如下,显示的为一个节点的实时状态图. 如图13 所示.

图13 运行截图

# 测试结果

## 功能性测试结果

### 单元测试结果

测试结果有bug后进行调试,改进.

最终每个测试的结果均符合预期.

### 集成测试结果

模块内功能与模块间通讯有bug后进行调试,改进.

最终模块内的功能测试结果与模块间通讯的测试结果均符合预期.

### 确认测试

每个需求分析中的功能均测试完毕, 符合预期.

## 非功能性测试结果

### 性能测试结果

测试环境:1000m无线路由器,windows10操作系统,edge浏览器

测试结果:

1. 平均打开系统页面:2s内
2. 打开”查看说明”:0.2s内
3. 打开”配置管理”:0.2s内
4. 使用”配置管理”修改一台机器配置:1.5s内
5. 查看”实时状态”:1s内
6. 实时状态刷新时间:平均1s/次
7. 查看”历史状态”:1s内

### 可扩展性测试结果

测试环境:开了20个client,3个浏览器访问.

测试结果:所有功能正常.并发访问正常,继续添加机器节点正常.

### 灵活性测试结果

代码使用面向对象思想, 使用了springboot框架,很多提供服务部分都封装了接口.预计可提供外部访问接口，能够实现后期如手机应用、外部应用的调用。程序升级简单.未实测.